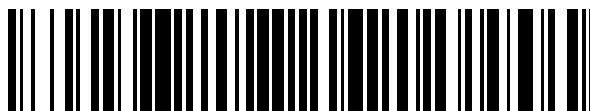


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 648 312**

51 Int. Cl.:

H04N 19/105 (2014.01)

H04N 19/61 (2014.01)

H04N 19/593 (2014.01)

H04N 19/182 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **12.06.2012 PCT/US2012/042052**

87 Fecha y número de publicación internacional: **20.12.2012 WO12173991**

96 Fecha de presentación y número de la solicitud europea: **12.06.2012 E 12728160 (8)**

97 Fecha y número de publicación de la concesión europea: **30.08.2017 EP 2719179**

54 Título: **Relleno de píxeles de borde para intra-predicción en codificación de vídeo**

30 Prioridad:

13.06.2011 US 201161496504 P

08.11.2011 US 201161557361 P

09.11.2011 US 201161557845 P

11.06.2012 US 201213493640

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

29.12.2017

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121, US

72 Inventor/es:

WANG, XIANGLIN;

CHIEN, WEI-JUNG;

KARCZEWICZ, MARTA;

CHEN, YING y

CHEN, PEISONG

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 648 312 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Relleno de píxeles de borde para intra-predicción en codificación de vídeo

5 **[0001]** Esta solicitud reivindica el beneficio de la solicitud provisional de Estados Unidos n.º 61/496 504, presentada el 13 de junio de 2011, la solicitud provisional de Estados Unidos n.º 61/557 361, presentada el 8 de noviembre de 2011 y la solicitud provisional de Estados Unidos n.º 61/557 845, presentada el 9 de noviembre de 2011.

CAMPO TÉCNICO

10 **[0002]** Esta divulgación se refiere a codificación de vídeo y, más particularmente, a intra-predicción de datos de vídeo.

ANTECEDENTES

15 **[0003]** Las capacidades de vídeo digital pueden incorporarse a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de difusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC) (H.264/AVC), la norma de Codificación de Vídeo de Alta eficiencia (HEVC), actualmente en desarrollo, y las extensiones de tales normas, para transmitir, recibir y almacenar información de vídeo digital de forma más eficaz.

25 **[0004]** Las técnicas de compresión de vídeo llevan a cabo la predicción espacial (intra-imagen) y/o la predicción temporal (inter-imagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un fragmento intra-codificado (I) de una imagen se codifican mediante predicción espacial con respecto a unas muestras de referencia de bloques próximos de la misma imagen. Los bloques de vídeo de un fragmento inter-codificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques próximos de la misma imagen, o la predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

30 **[0005]** Los documentos no patente VIKTOR WAHADANIAH ET AL: "Esquema de intra-predicción restringida para unidades de predicción de tamaño flexible en HEVC", 4. REUNIÓN DE JCT-VC; 95 REUNIÓN de MPEG; DAEGU (EQUIPO CONJUNTO DE COLABORACIÓN SOBRE CODIFICACIÓN DE VÍDEO DE LA ISO/ IEC JTC1/SC29/WG11AND ITU-T SG.16), n.º JCTVC-D094 de 14 de enero de 2011 (en lo sucesivo denominada JCTVC-D094) y RICKARD SJOBERG ET AL: "Restricción en la implementación del código fuente de intra", 4. REUNIÓN DE JCT-VC; 95 REUNIÓN MPEG; DAEGU (EQUIPO CONJUNTO DE COLABORACIÓN SOBRE CODIFICACIÓN DE VÍDEO DE LA ISO/IEC JTC1/SC29/WG11AND ITU-T SG.16), n.º JCTVC-D386 de 16 de enero de 2011 (en lo sucesivo denominada JCTVC-D386) divulgan un modo de INTRA-predicción restringido para HEVC que implica operaciones de relleno de píxeles de borde.

RESUMEN

50 **[0006]** En general, esta divulgación describe técnicas para relleno de píxeles de borde para intra-predicción en codificación de vídeo. De acuerdo con las técnicas de esta divulgación, el codificador de vídeo puede realizar una operación de relleno que asigna valores a píxeles de borde no disponibles. La operación de relleno puede procesar los píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede secuencialmente a través de los píxeles de borde hasta un píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno puede asignar un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado mediante la operación de relleno. El codificador de vídeo puede generar un bloque de vídeo intra-predicho basado en los píxeles de borde.

60 **[0007]** En un aspecto, esta divulgación describe un procedimiento para codificar datos de vídeo. El procedimiento comprende realizar una operación de relleno que procesa un conjunto de píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado por la operación de relleno. El procedimiento también comprende generar un bloque de vídeo intra-predicho basado en los píxeles de borde.

65 **[0008]** Esta divulgación también describe un aparato de codificación de vídeo que comprende uno o más

procesadores configurados para realizar una operación de relleno que procesa un conjunto de píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado por la operación de relleno. El uno o más procesadores también están configurados para generar un bloque de vídeo intra-predicho basado en los píxeles de borde.

[0009] Además, la presente divulgación describe un aparato de codificación de vídeo que comprende medios para realizar una operación de relleno que procesa un conjunto de píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede a través de los píxeles de borde secuencialmente hasta el píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado por la operación de relleno. El aparato de codificación de vídeo también comprende medios para generar un bloque de vídeo intra-predicho basado en los píxeles de borde.

[0010] Esta divulgación también describe un producto de programa informático que comprende uno o más medios de almacenamiento legibles por ordenador que almacenan instrucciones ejecutables por ordenador que, al ejecutarse, hacen que uno o más procesadores lleven a cabo una operación de relleno que procesa un conjunto de píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede secuencialmente a través de los píxeles de borde hasta un píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel previamente procesado por la operación de relleno. Las instrucciones también hacen que el uno o más procesadores generen un bloque de vídeo intra-predicho basado en los píxeles de borde.

[0011] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y los dibujos, y a partir de las reivindicaciones.

30 BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0012]

35 La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación de vídeo que puede implementar las técnicas de esta divulgación.

La FIG. 2 es un diagrama conceptual que ilustra un ejemplo de bloque de vídeo y un conjunto de píxeles de borde asociados con el bloque de vídeo.

40 La FIG. 3A es un diagrama conceptual que ilustra ejemplos de modos de intra-predicción en la codificación de vídeo de alta eficiencia.

La FIG. 3B es un diagrama conceptual que indica otro ejemplo de asignación de números a diferentes modos de intra-predicción.

45 La FIG. 3C es un diagrama conceptual que ilustra ejemplos de modos de intra-predicción en H.264/AVC.

La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar las técnicas de esta divulgación.

50 La FIG. 5 es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo que puede implementar las técnicas de esta divulgación.

La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de operación de intra-predicción.

55 La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de operación de relleno.

La FIG. 8 es un diagrama conceptual que ilustra un orden en el que el ejemplo de operación de relleno de la FIG. 7 procede a través de píxeles de borde.

60 La FIG. 9 es un diagrama de flujo que ilustra otro ejemplo de operación de relleno.

La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de orden en el que la operación de relleno de la FIG. 9 procede a través de píxeles de borde.

65 La FIG. 11 es un diagrama de flujo que ilustra otro ejemplo de operación de relleno.

La FIG. 12 es un diagrama conceptual que ilustra un ejemplo de orden en el que la operación de relleno de la FIG. 11 procede a través de píxeles de borde.

5 La FIG. 13 es un diagrama de flujo que ilustra un ejemplo de operación para determinar si un píxel de borde está disponible.

DESCRIPCIÓN DETALLADA

10 **[0013]** Los dibujos adjuntos ilustran ejemplos. Los elementos indicados mediante números de referencia en los dibujos adjuntos corresponden a elementos indicados mediante números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", etc.) no necesariamente implican que los elementos tienen un orden particular. Más bien, tales palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo o similar tipo.

15 **[0014]** Un codificador de vídeo puede generar un bloque de vídeo intra-predicho basándose en un conjunto de píxeles que bordean el bloque de vídeo en los lados superior e izquierdo del bloque de vídeo. Los píxeles de borde en el lado izquierdo del bloque de vídeo pueden extenderse por debajo de la fila inferior de píxeles del bloque de vídeo y los píxeles de borde en el lado superior del bloque de vídeo pueden extenderse a la derecha de la columna de píxeles de la derecha del bloque de vídeo.

20 **[0015]** En algunos casos, uno o más de los píxeles de borde tal vez no estén disponibles. Por consiguiente, el codificador de vídeo puede realizar una operación de relleno que asigna valores a píxeles de borde no disponibles. La operación de relleno puede procesar los píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede secuencialmente a través de los píxeles de borde hasta un píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno puede asignar un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado mediante la operación de relleno. Después de realizar la operación de relleno, el codificador de vídeo puede generar el bloque de vídeo intra-predicho basado en los píxeles de borde. Si el codificador de vídeo es un codificador de vídeo, el codificador de vídeo puede generar datos de vídeo codificados basados en el bloque de vídeo intra-predicho. Por ejemplo, el codificador de vídeo puede generar datos residuales basados, al menos en parte, en el bloque de vídeo intra-predicho. Si el codificador de vídeo es un descodificador de vídeo, el codificador de vídeo puede generar datos de vídeo descodificados basados en el bloque de vídeo intra-predicho. Por ejemplo, el descodificador de vídeo puede reconstruir un bloque de vídeo descodificado basado al menos en parte en el bloque de vídeo intra-predicho y un bloque de vídeo residual.

25 **[0016]** La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación de vídeo de ejemplo 10 que puede utilizar las técnicas de esta divulgación. Tal como se describe en el presente documento, el término "codificador de vídeo" se refiere genéricamente a codificadores de vídeo, descodificadores de vídeo y codificadores - descodificadores (CÓDEC) combinados. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a codificación de vídeo y descodificación de vídeo. Los términos codificador, codificador, descodificador y CÓDEC pueden referirse a máquinas específicas diseñadas para la codificación (codificación y/o descodificación) de datos de vídeo consistentes con esta divulgación. Un dispositivo que realiza codificación de vídeo puede denominarse un dispositivo o aparato de codificación de vídeo.

30 **[0017]** Como se muestra en la FIG. 1, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 y un dispositivo de destino 14. El dispositivo de origen 12 genera datos de vídeo codificados. El dispositivo de destino 14 puede descodificar los datos de vídeo codificados generados por el dispositivo de origen 12. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, notebooks (por ejemplo, portátiles), ordenadores tipo tableta, descodificadores, teléfonos portátiles como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, ordenadores de coche o similares. En algunos ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

35 **[0018]** El dispositivo de destino 14 puede recibir datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal 16. El canal 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el canal 16 puede comprender un medio de comunicación que permite al dispositivo de origen 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. En este ejemplo, el dispositivo de origen 12 puede modular los datos de vídeo codificados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y puede transmitir los datos de vídeo modulados al dispositivo de destino 14. El medio de comunicación puede comprender un medio de comunicación inalámbrico o alámbrico, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet.

El medio de comunicación puede incluir routers, conmutadores, estaciones base u otros equipos que faciliten la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

5 **[0019]** En otro ejemplo, el canal 16 puede corresponder a un medio de almacenamiento que almacena los datos de vídeo codificados generados por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder al medio de almacenamiento a través del acceso al disco o el acceso a la tarjeta. El dispositivo de almacenamiento puede incluir una variedad de medios de almacenamiento de datos de acceso local, tales como discos Blu-ray, DVD, CD-ROM, memoria flash u otros medios adecuados de almacenamiento digital para almacenar datos de vídeo codificados. En un ejemplo adicional, el canal 16 puede incluir un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacene el vídeo codificado generado por el dispositivo de origen 12. En este ejemplo, el dispositivo de destino 14 puede acceder a datos de vídeo codificados almacenados en el servidor de archivos u otro dispositivo de almacenamiento intermedio a través de transmisión o descarga. El servidor de archivos puede ser un tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir los datos de vídeo codificados al dispositivo de destino 14. Entre los ejemplos de servidores de archivos se incluyen servidores web (por ejemplo, para un sitio web), servidores FTP, dispositivos de almacenamiento conectados a la red (NAS) y unidades de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de una conexión de datos estándar, como una conexión a Internet. Entre los ejemplos de tipos de conexiones de datos pueden incluirse canales inalámbricos (por ejemplo, conexiones Wi-Fi), conexiones alámbricas (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el servidor de archivos puede ser una transmisión sin descarga, una transmisión de descarga o una combinación de ambas.

15 **[0020]** Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones de multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede configurarse para soportar transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como transmisión de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

25 **[0021]** En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador / desmodulador (módem) y/o un transmisor. La fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene datos de vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir datos de vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de vídeo, o una combinación de dichas fuentes.

35 **[0022]** El codificador de vídeo 20 puede codificar datos de vídeo. La interfaz de salida 22 puede transmitir directamente los datos de vídeo codificados al dispositivo de destino 14. De forma alternativa, la interfaz de salida 22 puede almacenar los datos de vídeo codificados en un medio de almacenamiento o en un servidor de archivos para un acceso posterior mediante el dispositivo de destino 14 para la decodificación y/o reproducción.

40 **[0023]** En el ejemplo de la FIG. 1, el dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados por el canal 16. Los datos de vídeo codificados pueden incluir una variedad de elementos sintácticos que representan los datos de vídeo.

45 **[0024]** El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra datos de vídeo decodificados a un usuario. El dispositivo de visualización 32 puede comprender cualquiera entre una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

50 **[0025]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de elaboración, y pueden ajustarse a un modelo de prueba HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, también denominada MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Entre otros ejemplos de normas de compresión de vídeo se incluyen MPEG-2 e ITU-T H.263.

55 **[0026]** Aunque no se muestra en el ejemplo de la FIG. 1, el codificador de vídeo 20 y el decodificador de vídeo 30

pueden estar integrados, cada uno de ellos, con un codificador y un decodificador de audio, y pueden incluir unidades adecuadas MUX-DEMUX, u otro tipo de hardware y software, para gestionar la codificación, tanto de audio como de vídeo, en un flujo de datos común o en flujos de datos diferentes. En algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0027] Una vez más, la FIG. 1 es meramente un ejemplo y las técnicas de esta divulgación pueden aplicarse a ajustes de codificación de vídeo (por ejemplo, codificación de vídeo o decodificación de vídeo) que no incluyen necesariamente ninguna comunicación de datos entre dispositivos de codificación y decodificación. En muchos ejemplos, la codificación de vídeo y la decodificación de vídeo se realiza mediante dispositivos que no se comunican entre sí, sino que almacenan datos de vídeo codificados en la memoria y/o recuperan y decodifican datos de vídeo codificados de la memoria.

[0028] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse cada uno utilizando varios tipos de circuitería, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, hardware o cualquier combinación de estos. En algunos casos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse al menos parcialmente en software. Cuando el codificador de vídeo 20 y/o el decodificador de vídeo 30 se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en unos medios de almacenamiento legibles por ordenador no transitorios adecuados, y puede ejecutar las instrucciones en hardware, usando uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o decodificadores, donde cualquiera de ambos puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

[0029] Como se ha mencionado brevemente anteriormente, el codificador de vídeo 20 codifica los datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, una imagen puede denominarse una "trama" de vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forma una representación codificada de los datos de vídeo.

[0030] Para generar el flujo de bits, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada imagen en los datos de vídeo. Cuando el codificador de vídeo 20 realiza operaciones de codificación en las imágenes, el codificador de vídeo 20 puede generar una serie de imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen. Los datos asociados pueden incluir conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen, conjuntos de parámetros de adaptación y otras estructuras sintácticas. Un conjunto de parámetros de secuencia (SPS) puede contener parámetros aplicables a cero o más secuencias de imágenes. Un conjunto de parámetros de imagen (PPS) puede contener parámetros aplicables a cero o más imágenes. Un conjunto de parámetros de adaptación (APS) puede contener parámetros aplicables a cero o más imágenes. Los parámetros en el PPS pueden fijarse para un conjunto de imágenes, mientras que los parámetros en diferentes conjuntos de parámetros de adaptación pueden ser aplicables a diferentes imágenes del conjunto de imágenes.

[0031] Para generar una imagen codificada, el codificador de vídeo 20 puede dividir una imagen en bloques de vídeo de igual tamaño que no se superpongan. Cada uno de los bloques de vídeo está asociado con un bloque de árbol. En algunos casos, un bloque de árbol también puede denominarse una unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser aproximadamente análogos a los macrobloques de normas anteriores, tales como la H.264/AVC. Sin embargo, un bloque de árbol no está limitado necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar la división en árbol cuádruple para dividir los bloques de vídeo de bloques de árbol en bloques de vídeo asociados con las CU, de ahí el nombre "bloques de árbol".

[0032] En algunos ejemplos, el codificador de vídeo 20 puede dividir una imagen en una pluralidad de fragmentos. Cada uno de los fragmentos puede incluir un número entero de CU. En algunos casos, un fragmento comprende un número entero de bloques de árbol. En otros casos, un límite de un fragmento puede estar dentro de un bloque de árbol.

[0033] Como parte de la realización de una operación de codificación en una imagen, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada fragmento de la imagen. Cuando el codificador de vídeo 20 realiza una operación de codificación en un fragmento, el codificador de vídeo 20 puede generar datos codificados asociados con el fragmento. Los datos codificados asociados con el fragmento pueden denominarse "fragmento codificado".

[0034] Para generar un fragmento codificado, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada bloque de árbol en un fragmento. Cuando el codificador de vídeo 20 realiza una operación de codificación en un bloque de árbol, el codificador de vídeo 20 puede generar un bloque de árbol codificado. El

bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.

5 **[0035]** Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede llevar a cabo de forma recursiva división en árbol cuádruple en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado con una CU diferente. Por ejemplo, el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol en cuatro sub-bloques de igual tamaño, dividir uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, etc. Uno o más elementos sintácticos en el flujo de bits puede indicar un número máximo de veces que el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol. Un bloque de vídeo de una CU puede tener forma cuadrada. El tamaño del bloque de vídeo de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de vídeo de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o mayor

15 **[0036]** El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de árbol. Una CU no dividida es una CU cuyo bloque de vídeo no está dividido en bloques de vídeo para otras CU. Como parte de realizar una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU puede estar asociada con un bloque de vídeo diferente dentro del bloque de vídeo de la CU. El codificador de vídeo 20 puede generar un bloque de vídeo predicho para cada PU de la CU. El bloque de vídeo predicho de una PU puede ser un bloque de muestras. En esta divulgación, el término "muestra" puede referirse a un brillo (luminancia) o color (crominancia) de un píxel. El codificador de vídeo 20 puede usar intra-predicción o inter-predicción para generar el bloque de vídeo predicho para una PU.

20 **[0037]** Cuando el codificador de vídeo 20 utiliza intra-predicción para generar el bloque de vídeo predicho de una PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basado en muestras de la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la intra-predicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intra-predicha.

25 **[0038]** Cuando el codificador de vídeo 20 utiliza inter-predicción para generar el bloque de vídeo predicho de la PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basado en valores de imágenes distintas de la imagen asociada con la PU. En otras palabras, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basado en muestras en una o más imágenes de referencia. Además, cuando el codificador de vídeo 20 utiliza inter-predicción para generar un bloque de vídeo predicho para una PU, el codificador de vídeo 20 puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar una porción de una imagen de referencia que corresponde al bloque de vídeo de la PU. En otras palabras, la información de movimiento para una PU puede indicar una "muestra de referencia" para la PU. El codificador de vídeo 20 puede generar el bloque de vídeo predicho para la PU basándose en las porciones de las imágenes de referencia que están indicadas por la información de movimiento para la PU. Si el codificador de vídeo 20 utiliza la inter-predicción para generar bloques de vídeo predichos para las PU de una CU, la CU es una CU inter-predicha.

30 **[0039]** Después de que el codificador de vídeo 20 genere bloques de vídeo predichos para una o más PU de una CU, el codificador de vídeo 20 puede generar datos residuales para la CU basándose en los bloques de vídeo predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de vídeo predichos para las PU de la CU y el bloque de vídeo original de la CU.

35 **[0040]** Además, como parte de la realización de una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede realizar la división en árbol cuádruple recursiva en los datos residuales de la CU para dividir los datos residuales de la CU en uno o más bloques de datos residuales (es decir, bloques de vídeo residuales) asociados con unidades de transformación (TU) de la CU. Cada TU de una CU puede estar asociada con un bloque de vídeo residual diferente. El codificador de vídeo 20 puede realizar operaciones de transformación en cada TU de la CU.

40 **[0041]** Cuando el codificador de vídeo 20 realiza la operación de transformación en una TU, el codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de vídeo residual asociado con la TU para generar uno o más bloques de coeficientes de transformación (es decir, bloques de coeficientes de transformación) asociados con la TU. Conceptualmente, un bloque de coeficientes de transformación puede ser una matriz bidimensional (2D) de coeficientes de transformación.

45 **[0042]** Después de generar un bloque de coeficientes de transformación, el codificador de vídeo 20 puede realizar una operación de cuantificación en el bloque de coeficientes de transformación. La cuantificación se refiere en general a un proceso en el que los coeficientes de transformación se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. La cuantificación puede reducir la profundidad de bits asociada con algunos o la totalidad de los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de n bits puede redondearse a la baja hasta un coeficiente de transformación de m bits durante la cuantificación, donde n es mayor que m .

[0043] El codificador de vídeo 20 puede asociar cada CU o un número de CU con un valor de parámetro de cuantificación (QP). El valor QP asociado con una CU puede determinar cómo el codificador de vídeo 20 cuantifica los bloques de coeficientes de transformación asociados con la CU o el número de CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con una CU ajustando el valor QP asociado con la CU.

[0044] Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes de transformación, el codificador de vídeo 20 puede escanear los coeficientes de transformación cuantificados para producir un vector unidimensional de niveles de coeficientes de transformación. El codificador de vídeo 20 puede codificar por entropía el vector unidimensional. El codificador de vídeo 20 también puede codificar por entropía otros elementos sintácticos asociados con los datos de vídeo.

[0045] El flujo de bits generado por el codificador de vídeo 20 puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura sintáctica que contiene una indicación de un tipo de datos en la unidad NAL y los bytes que contienen los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un fragmento codificado, información de mejora suplementaria (SEI), un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos en una unidad NAL pueden incluir estructuras sintácticas codificadas por entropía, tales como bloques de coeficientes de transformación codificados por entropía, información de movimiento, etc.

[0046] El descodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. El flujo de bits puede incluir una representación codificada de los datos de vídeo codificados por el codificador de vídeo 20. Cuando el descodificador de vídeo 30 recibe el flujo de bits, el descodificador de vídeo 30 puede realizar una operación de análisis sintáctico en el flujo de bits. Cuando el descodificador de vídeo 30 realiza la operación de análisis sintáctico, el descodificador de vídeo 30 puede extraer elementos sintácticos del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose en los elementos sintácticos extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo basados en los elementos sintácticos puede ser en general recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos sintácticos.

[0047] Después de que el descodificador de vídeo 30 extraiga los elementos sintácticos asociados con una CU, el descodificador de vídeo 30 puede generar bloques de vídeo intra-predicho o inter-predicho para las PU de la CU basándose en los elementos sintácticos. Además, el descodificador de vídeo 30 puede invertir bloques de coeficientes de transformación cuantificados asociados con TU de la CU. El descodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes de transformación para reconstruir los bloques de vídeo residuales asociados con las TU de la CU. Después de generar los bloques de vídeo predichos y reconstruir los bloques de vídeo residuales, el descodificador de vídeo 30 puede reconstruir el bloque de vídeo de la CU basado en los bloques de vídeo predichos y los bloques de vídeo residuales. De esta manera, el descodificador de vídeo 30 puede determinar los bloques de vídeo de las CU basándose en los elementos sintácticos en el flujo de bits.

[0048] Como se ha descrito brevemente más arriba, un codificador de vídeo, como el codificador de vídeo 20 o el descodificador de vídeo 30, puede utilizar intra-predicción para generar un bloque de vídeo predicho para una PU. Para facilitar la explicación, esta divulgación puede referirse a un bloque de vídeo predicho generado usando intra-predicción como un bloque de vídeo intra-predicho. Cuando el codificador de vídeo genera un bloque de vídeo intra-predicho para una PU, el codificador de vídeo puede asignar los valores de varios píxeles de borde a píxeles en el bloque de vídeo intra-predicho.

[0049] La FIG. 2 es un diagrama conceptual que ilustra un bloque de vídeo de ejemplo 50 y un conjunto de píxeles de borde asociados con el bloque de vídeo 50. El bloque de vídeo 50 puede estar asociado con una PU de una CU que un codificador de vídeo está codificando actualmente. En el ejemplo de la FIG. 2, el bloque de vídeo 50 consta de dieciséis píxeles denominados a-p. En el ejemplo de la FIG. 2, los píxeles de borde se denominan A-R. Otros bloques de vídeo pueden ser mayores o menores que el bloque de vídeo 50. En general, un bloque de vídeo puede estar asociado con $2 \cdot w + 1$ píxeles de borde por encima del bloque de vídeo y $2 \cdot h + 1$ píxeles de borde a la izquierda del bloque de vídeo, donde w es la anchura y h es la altura del bloque de vídeo.

[0050] Los píxeles de borde por encima de un bloque de vídeo (por ejemplo, bloque de vídeo 50) pueden formar un predictor superior. En otras palabras, el predictor superior puede ser una matriz de muestras de referencia correspondientes a una fila de muestras situadas por encima del bloque de vídeo. Si el píxel superior izquierdo del bloque de vídeo tiene coordenadas (x, y) y el tamaño del bloque de vídeo es $N \times N$, las muestras con coordenadas $(x+i, y-1)$, donde i varía de -1 a $2N$, forman el predictor superior. Los píxeles de borde a la izquierda del bloque de vídeo forman un predictor izquierdo. En otras palabras, el predictor izquierdo puede ser una matriz de muestras de referencia correspondientes a una columna de muestras situada a la izquierda del bloque de vídeo. Si el píxel superior izquierdo del bloque de vídeo tiene coordenadas (x, y) y el tamaño del bloque de vídeo es $N \times N$, las muestras con coordenadas $(x-1, y+j)$, donde j varía de -1 a $2N$, forman el predictor izquierdo.

[0051] La FIG. 3A es un diagrama conceptual que ilustra ejemplos de modos de intra-predicción en HEVC. En el ejemplo de la FIG. 3A, los modos de intra-predicción 1, 2 y 4-33 corresponden a diferentes direcciones a partir de las cuales un codificador de vídeo puede extender píxeles de borde para generar un bloque de vídeo intra-predicho. Por ejemplo, cuando el codificador de vídeo usa el modo de intra-predicción 1, el codificador de vídeo puede generar un bloque de vídeo intra-predicho extendiendo los valores de los píxeles de borde verticalmente hacia abajo. En este ejemplo, el codificador de vídeo puede usar el modo de intra-predicción 1 para generar el bloque de vídeo 50 (FIG. 2) asignando el valor de píxel de borde A a los píxeles a, e, i y m, asignando el valor de píxel de borde B a los píxeles b, f, j y n, asignando el valor de píxel de borde C a los píxeles c, g, k y o, y asignando el valor de píxel de borde D a los píxeles d, h, l, y p.

[0052] El modo 0 puede referirse a un modo planar. El codificador de vídeo puede usar la siguiente fórmula para generar un bloque de vídeo intra-predicho usando el modo planar:

$$\text{predSamples}[x, y] = ((nS - 1 - x) * p[-1, y] + (x + 1) * p[nS, -1] + (nS - 1 - y) * p[x, -1] + (y + 1) * p[-1, nS] + nS) \gg (k + 1)$$

con $x, y = 0..nS-1$ donde $k = \text{Log}_2(nS)$.

En la fórmula anterior, $\text{predSamples}[x, y]$ es una matriz de muestras intra-predichas, nS indica un tamaño del bloque de vídeo predicho, y $p[x, y]$ es una matriz de muestras próximas.

[0053] El modo 3 puede referirse a un modo DC. Para generar un bloque de vídeo intra-predicho usando el modo DC, el codificador de vídeo puede calcular primero un valor $DCVal$ usando la siguiente fórmula:

$$DCVal = \left(\sum_{x'=0}^{nS-1} p[x', -1] + \sum_{y'=0}^{nS-1} p[-1, y'] + nS \right) \gg (k + 1),$$

con $x, y = 0..nS-1$ Si un índice de crominancia ($cldx$) de un bloque actual es igual a 0, se aplica lo siguiente.

$$\text{predSamples}[0, 0] = (1 * p[-1, 0] + 2 * DCVal + 1 * p[0, -1] + 2) \gg 2$$

$$\text{predSamples}[x, 0] = (1 * p[x, -1] + 3 * DCVal + 2) \gg 2,$$

con $x = 1..nS-1$

$$\text{predSamples}[0, y] = (1 * p[-1, y] + 3 * DCVal + 2) \gg 2,$$

con $y = 1..nS-1$

$$\text{predSamples}[x, y] = DCVal,$$

con $x, y = 1..nS-1$

De lo contrario, las muestras de predicción $\text{predSamples}[x, y]$ se obtienen como

$$\text{predSamples}[x, y] = DCVal,$$

con $x, y = 0..nS-1$

En las fórmulas anteriores, $\text{predSamples}[x, y]$ es una matriz de muestras intra-predichas, nS indica un tamaño del bloque de vídeo predicho, y $p[x, y]$ es una matriz de muestras próximas.

[0054] En otro ejemplo, cuando el codificador de vídeo utiliza el modo de intra-predicción 10, el codificador de vídeo puede generar un bloque de vídeo intra-predicho mediante la extensión de los valores de píxeles de borde en diagonal desde la parte inferior izquierda a la parte superior derecha. En el ejemplo de la FIG. 2, el codificador de vídeo puede utilizar el modo de intra-predicción 10 para generar el bloque de vídeo 50 asignando el valor de píxel de borde J al píxel a, asignando el valor de píxel de borde K a los píxeles e y b, asignando el valor de píxel de borde L a los píxeles i, f y c, asignando el valor de píxel de borde M a los píxeles m, j, g y d, asignando el valor de píxel de borde N a los píxeles n, k y h, asignando el valor de píxel de borde O a los píxeles o y l, y asignando el valor de píxel de borde P al píxel p.

[0055] En otros ejemplos, el codificador de vídeo puede utilizar diferentes números para indicar los modos de intra-

predicción. Por ejemplo, la FIG. 3B es un diagrama conceptual que indica otra asignación de números a diferentes modos de intra-predicción.

5 **[0056]** La FIG. 3C es un diagrama conceptual que ilustra ejemplos de modos de intra-predicción en H.264/AVC. Puede haber menos modos de intra-predicción disponibles en H.264/AVC.

10 **[0057]** En algunos casos, uno o más de los píxeles de borde asociados con un bloque de vídeo intra-predicho no están disponibles. El conjunto de píxeles de borde asociados con un bloque de vídeo intra-predicho puede incluir los píxeles de borde que el codificador de vídeo puede utilizar para generar el bloque de vídeo intra-predicho. Cuando un píxel de borde no está disponible, el codificador de vídeo tal vez no pueda acceder al valor de píxel de borde. Por consiguiente, el codificador de vídeo puede ser incapaz de usar el valor de píxel de borde para generar un bloque de vídeo intra-predicho.

15 **[0058]** Un píxel de borde tal vez no esté disponible por una variedad de razones. Por ejemplo, un píxel de borde tal vez no esté disponible si el píxel de borde está fuera del borde de la imagen actual (es decir, la imagen que el codificador de vídeo está codificando actualmente). En otro ejemplo, un píxel de borde tal vez no esté disponible si el píxel de borde está fuera del borde de un fragmento o elemento actual (es decir, el fragmento o elemento que el codificador de vídeo está codificando actualmente) y no está habilitada la predicción de fragmento / elemento cruzado. En otro ejemplo, un píxel de borde tal vez no esté disponible si el píxel de borde está asociado con una CU inter-predicha y la intra-predicción restringida está habilitada para la imagen actual.

20 **[0059]** Debido a que uno o más píxeles de borde tal vez no estén disponibles, el codificador de vídeo puede llevar a cabo una operación de relleno que asigna valores a píxeles de borde no disponibles. Después de que el codificador de vídeo realice la operación de relleno, el codificador de vídeo puede utilizar los valores disponibles de los píxeles de borde disponibles y los valores asignados de los píxeles de borde no disponibles para generar un bloque de vídeo intra-predicho. Cada uno de los documentos que no son de patente citados anteriormente JCTVC-D094 y JCTVC-D386 divulga ejemplos de tales operaciones de relleno.

30 **[0060]** Los diferentes codificadores de vídeo pueden asignar valores a los píxeles de borde no disponibles de varias maneras. Por ejemplo, el codificador de vídeo puede realizar una operación de relleno que escanea a través de los píxeles de borde desde un píxel de borde inferior izquierdo (por ejemplo, el píxel de borde P en la FIG. 2) hasta un píxel de esquina (por ejemplo, el píxel de borde R en la FIG. 2) hasta un píxel de borde superior derecho (por ejemplo, el píxel de borde H en la FIG. 2). En este ejemplo, cuando el codificador de vídeo alcanza un píxel de borde no disponible, el codificador de vídeo puede escanear hacia adelante hasta que el codificador de vídeo alcance un siguiente píxel de borde disponible. Al identificar el píxel de borde disponible, la operación de relleno puede asignar un valor al píxel de borde no disponible basándose en los valores del siguiente píxel disponible y un píxel de borde disponible anterior.

40 **[0061]** En el ejemplo de operación de relleno descrito anteriormente, si no hay ningún píxel de borde disponible anterior, el codificador de vídeo puede escanear por delante para encontrar el primer píxel de borde disponible y a continuación asignar el valor del primer píxel de borde disponible a cada píxel de borde no disponible antes del primer píxel de borde disponible. En otras palabras, $P_{\text{pad}} = P_{\text{next}}$, donde P_{pad} es el valor asignado a cada píxel de borde no disponible antes del primer píxel de borde disponible y P_{next} es el valor del siguiente píxel de borde disponible. Si no hay ningún píxel de borde disponible siguiente, el codificador de vídeo puede extrapolar el valor del píxel de borde no disponible a partir del valor del último píxel de borde disponible. En otras palabras, $P_{\text{pad}} = P_{\text{pre}}$, donde P_{pad} es el valor de cada píxel de borde no disponible después del último píxel de borde disponible y P_{pre} es el valor del último píxel de borde disponible. Si no se dispone de píxeles de borde de P_{pre} y P_{next} , el codificador de vídeo puede asignar un valor predeterminado a los píxeles de borde.

50 **[0062]** La asignación de valores a los píxeles de borde no disponibles en la forma descrita en los párrafos anteriores puede tener varias desventajas. Por ejemplo, asignar valores a cada píxel de borde no disponible de esta manera puede requerir que el codificador de vídeo realice varias operaciones de lectura de memoria para escanear hacia adelante para encontrar el siguiente píxel de borde disponible. En consecuencia, la operación de relleno descrita anteriormente puede requerir un acceso de memoria de estilo saliente. Por ejemplo, una vez que el codificador de vídeo localiza un siguiente píxel de borde disponible, el codificador de vídeo puede asignar un valor P_{pad} a un píxel de borde no disponible. A continuación, el codificador de vídeo puede asignar el valor P_{pad} a cada píxel de borde desde un píxel de borde actual hasta el siguiente píxel de borde disponible, lo cual puede requerir accesos de memoria empezando de nuevo desde el píxel de borde actual. Tal patrón de acceso de memoria de estilo de vaivén puede no ser deseable. Las operaciones de acceso a memoria pueden consumir mucho tiempo y consumir mucha energía, especialmente cuando se accede a la memoria de forma aleatoria en términos de ubicación de datos.

60 **[0063]** De acuerdo con las técnicas de la presente divulgación, el codificador de vídeo puede realizar una operación de relleno que procesa un conjunto de píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo (por ejemplo, el píxel de borde P en la FIG. 2) y procede a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho (por ejemplo, el píxel de borde H en la FIG. 2). Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de

borde no disponible basándose en un valor de un píxel de borde previamente procesado por la operación de relleno. De esta manera, el codificador de vídeo no necesita escanear hacia delante para encontrar el siguiente píxel de borde disponible. Al mismo tiempo, esta operación de relleno puede tener poco impacto en la eficiencia de codificación de los datos de vídeo. Debido a que el codificador de vídeo puede usar un patrón de acceso a memoria más regular y consistente, puede reducirse la complejidad del codificador de vídeo y mejorarse el rendimiento del codificador de vídeo en relación con los codificadores de vídeo que implementan otras técnicas.

[0064] La FIG. 4 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que está configurado para implementar las técnicas de esta divulgación. La FIG. 4 se proporciona con fines de explicación y no debe considerarse limitante de las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

[0065] En el ejemplo de la FIG. 4, el codificador de vídeo 20 incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador de vídeo 20 incluyen un módulo de predicción 100, un módulo de generación residual 102, un módulo de transformación 104, un módulo de cuantificación 106, un módulo de cuantificación inversa 108, un módulo de transformación inversa 110, un módulo de reconstrucción 112, un módulo de filtro 113, una memoria intermedia de imágenes descodificadas 114, y un módulo de codificación por entropía 116. El módulo de predicción 100 incluye un módulo de estimación de movimiento 122, un módulo de compensación de movimiento 124 y un módulo de intra-predicción 126. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales. Además, el módulo de estimación de movimiento 122 y el módulo de compensación de movimiento 124 pueden estar altamente integrados, pero están representados en el ejemplo de la FIG. 4 por separado con fines de explicación.

[0066] El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede recibir los datos de vídeo de diversas fuentes. Por ejemplo, el codificador de vídeo 20 puede recibir los datos de vídeo de la fuente de vídeo 18 (FIG. 1) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador de vídeo 20 puede realizar una operación de codificación en cada una de las imágenes. Como parte de realizar la operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Como parte de realizar una operación de codificación en un fragmento, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el fragmento.

[0067] Como parte de la realización de una operación de codificación en un bloque de árbol, el módulo de predicción 100 puede realizar la división en árbol cuádruple en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado con una CU diferente. Por ejemplo, el módulo de predicción 100 puede dividir un bloque de vídeo de un bloque de árbol en cuatro sub-bloques de igual tamaño, dividir uno o más de los sub-bloques en cuatro sub-sub-bloques de igual tamaño, etc.

[0068] Los tamaños de los bloques de vídeo asociados con las CU pueden variar desde 8x8 muestras hasta el tamaño del bloque de árbol, con un máximo de 64x64 muestras o más. En esta divulgación, "NxN" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de muestras de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 muestras o 16 por 16 muestras. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en una dirección vertical ($y = 16$) y dieciséis muestras en una dirección horizontal ($x = 16$). Asimismo, un bloque NxN presenta en general N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

[0069] Además, como parte de la realización de la operación de codificación en un bloque de árbol, el módulo de predicción 100 puede generar una estructura de datos jerárquica en árbol cuádruple para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos en árbol cuádruple. Si el módulo de predicción 100 divide el bloque de vídeo del bloque de árbol en cuatro sub-bloques, el nodo raíz tiene cuatro nodos secundarios en la estructura de datos en árbol cuádruple. Cada uno de los nodos secundarios corresponde a una CU asociada con uno de los sub-bloques. Si el módulo de predicción 100 divide uno de los sub-bloques en cuatro sub-sub-bloques, el nodo correspondiente a la CU asociada con el sub-bloque puede tener cuatro nodos secundarios, cada uno de los cuales correspondiente a una CU asociada con uno de los sub-sub-bloques, etc.

[0070] Cada nodo de la estructura de datos en árbol cuádruple puede contener datos sintácticos (por ejemplo, elementos sintácticos) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si el bloque de vídeo de la CU correspondiente al nodo está dividido (es decir, partido) en cuatro sub-bloques. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si el bloque de vídeo de la CU está dividido en sub-bloques. Una CU cuyo bloque de vídeo no está dividido puede corresponder a un nodo hoja en la estructura de datos en árbol cuádruple. Un bloque de árbol codificado puede incluir datos basados en la estructura de datos en árbol cuádruple para un bloque de árbol correspondiente.

[0071] El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de árbol. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede generar datos que representan una representación codificada de la CU no dividida.

[0072] Como parte de la realización de una operación de codificación en una CU, el módulo de predicción 100 puede dividir el bloque de vídeo de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden soportar varios tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden soportar intra-predicción en tamaños de PU de $2N \times 2N$ o $N \times N$ e inter-predicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$ o similares. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden soportar divisiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En algunos ejemplos, el módulo de predicción 100 puede realizar una división geométrica para dividir el bloque de vídeo de una CU entre PU de la CU a lo largo de un límite que no coincide con los lados del bloque de vídeo de la CU en ángulos rectos.

[0073] El módulo de estimación de movimiento 122 y el módulo de compensación de movimiento 124 pueden realizar inter-predicción en cada PU de la CU. La inter-predicción puede proporcionar compresión temporal. Mediante la realización de inter-predicción en una PU, el módulo de estimación de movimiento 122 y el módulo de compensación de movimiento 124 pueden generar datos de predicción para la PU basados en muestras descodificadas de imágenes de referencia distintas de la imagen asociada con la CU. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y varios elementos sintácticos.

[0074] Por otra parte, cuando el módulo de estimación de movimiento 122 realiza una operación de estimación de movimiento con respecto a una PU, el módulo de estimación de movimiento 122 puede generar uno o más vectores de movimiento para la PU. Por ejemplo, los fragmentos pueden ser fragmentos I, fragmentos P o fragmentos B. El módulo de estimación de movimiento 122 y el módulo de compensación de movimiento 124 pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento en I, todas las PU son intra-predichas. Por lo tanto, si la PU está en un fragmento I, el módulo de estimación de movimiento 122 y el módulo de compensación de movimiento 124 no realizan inter-predicción en la PU.

[0075] Si la PU está en un fragmento P, la imagen que contiene la PU está asociada con una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia en la lista 0 contiene muestras que pueden usarse para la inter-predicción de imágenes posteriores en orden de descodificación. Cuando el módulo de estimación de movimiento 122 realiza la operación de estimación de movimiento con respecto a una PU en un fragmento P, el módulo de estimación de movimiento 122 puede buscar las imágenes de referencia en la lista 0 para una muestra de referencia para la PU. La muestra de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras, que se corresponde más estrechamente con las muestras en el bloque de vídeo de la PU. El módulo de estimación de movimiento 122 puede usar una variedad de métricas para determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras en el bloque de vídeo de una PU. Por ejemplo, el módulo de estimación de movimiento 122 puede determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras en el bloque de vídeo de una PU por la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otra diferencia métrica.

[0076] Después de identificar una muestra de referencia de una PU en un fragmento P, el módulo de estimación de movimiento 122 puede generar un índice de referencia que indica la imagen de referencia de la lista 0 que contiene la muestra de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y la muestra de referencia. En varios ejemplos, el módulo de estimación de movimiento 122 puede generar vectores de movimiento con diferentes grados de precisión. Por ejemplo, el módulo de estimación de movimiento 122 puede generar vectores de movimiento con un cuarto de precisión de muestra, un octavo de precisión de muestra u otra fracción de precisión de muestra. En el caso de la precisión de la muestra fraccionada, los valores de las muestras de referencia pueden interpolarse a partir de valores de muestra de posición de entero en la imagen de referencia. El módulo de estimación de movimiento 122 puede enviar información de movimiento para la PU al módulo de codificación por entropía 116 y al módulo de compensación de movimiento 124. La información de movimiento para la PU puede incluir el índice de referencia y el vector de movimiento de la PU. El módulo de compensación de movimiento 124 puede utilizar la información de movimiento de una PU para identificar y recuperar la muestra de referencia de la PU.

[0077] Si la PU está en un fragmento B, la imagen que contiene la PU puede estar asociada con dos listas de imágenes de referencia, denominadas "lista 0" y "lista 1." Cada una de las imágenes de referencia en la lista 0 contiene muestras que pueden usarse para la inter-predicción de imágenes posteriores en orden de descodificación. Las imágenes de referencia de la lista 1 se producen antes de la imagen en orden de descodificación pero después de la imagen en orden de presentación. En algunos ejemplos, una imagen que contiene un fragmento B puede estar asociada con una combinación de listas que es una combinación de la lista 0 y la lista 1.

[0078] Además, si la PU está en un fragmento B, el módulo de estimación de movimiento 122 puede llevar a cabo predicción unidireccional o predicción bidireccional para la PU. Cuando el módulo de estimación de movimiento 122 realiza la predicción unidireccional para la PU, el módulo de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 o la lista 1 para una muestra de referencia para la PU. El módulo de estimación de movimiento 122 puede entonces generar un índice de referencia que indica la imagen de referencia en la lista 0 o la lista 1 que contiene la muestra de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y la muestra de referencia. El módulo de estimación de movimiento 122 puede enviar elementos sintácticos que indican la información de movimiento para la PU al módulo de codificación por entropía 116 y el módulo de compensación de movimiento 124. La información de movimiento para la PU puede incluir el índice de referencia, un indicador de dirección de predicción y el vector de movimiento del PU. El indicador de dirección de predicción puede indicar si el índice de referencia indica una imagen de referencia en la lista 0 o la lista 1. El módulo de compensación de movimiento 124 puede usar la información de movimiento de la PU para identificar y recuperar la muestra de referencia de la PU.

[0079] Cuando el módulo de estimación de movimiento 122 realiza la predicción bidireccional para una PU, el módulo de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 para una muestra de referencia para la PU y también puede buscar las imágenes de referencia de la lista 1 para otra muestra de referencia para la PU. El módulo de estimación de movimiento 122 puede entonces generar índices de referencia que indican las imágenes de referencia en la lista 0 y la lista 1 que contienen las muestras de referencia y los vectores de movimiento que indican desplazamientos espaciales entre las muestras de referencia y la PU. El módulo de estimación de movimiento 122 puede enviar elementos sintácticos que indican la información de movimiento de la PU al módulo de codificación por entropía 116 y el módulo de compensación de movimiento 124. La información de movimiento para la PU puede incluir los índices de referencia y los vectores de movimiento de la PU. El módulo de compensación de movimiento 124 puede utilizar la información de movimiento para identificar y recuperar la muestra de referencia de la PU.

[0080] En algunos casos, el módulo de estimación de movimiento 122 no envía un conjunto completo de información de movimiento para una PU al módulo de codificación por entropía 116. En lugar de eso, el módulo de estimación de movimiento 122 puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, el módulo de estimación de movimiento 122 puede determinar que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU próxima. En este ejemplo, el módulo de estimación de movimiento 122 puede indicar, en un nodo en árbol cuádruple para una CU asociada con la PU, un valor que indica al descodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU próxima. En otro ejemplo, el módulo de estimación de movimiento 122 puede identificar, en un nodo en árbol cuádruple asociado con la CU asociada con la PU, una PU próxima y una diferencia de vectores de movimiento (MVD). La diferencia de vectores de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU próxima indicada. El descodificador de vídeo 30 puede utilizar el vector de movimiento de la PU próxima indicada y la diferencia de vectores de movimiento para predecir el vector de movimiento de la PU. Haciendo referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador de vídeo 20 puede ser capaz de señalar la información de movimiento de la segunda PU usando menos bits.

[0081] Como parte de la realización de una operación de codificación en una CU, el módulo de intra-predicción 126 puede llevar a cabo la intra-predicción en las PU de la CU. La intra-predicción puede proporcionar compresión espacial. Cuando el módulo de intra-predicción 126 realiza intra-predicción en una PU, el módulo de intra-predicción 126 puede generar datos de predicción para la PU basados en muestras descodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y varios elementos sintácticos. El módulo de intra-predicción 126 puede realizar intra-predicción en PU en fragmentos I, fragmentos P y fragmentos B.

[0082] Para realizar intra-predicción en una PU, el módulo de intra-predicción 126 puede usar múltiples modos de intra-predicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando el módulo de intra-predicción 126 utiliza un modo de intra-predicción para generar un conjunto de datos de predicción para la PU, el módulo de intra-predicción 126 puede extender muestras de bloques de vídeo de PU próximas a través del bloque de vídeo de la PU en una dirección asociada con el modo de intra-predicción. Las PU próximas pueden estar por encima, por encima y a la derecha, por encima y a la izquierda, a la izquierda, o por debajo y a la izquierda de la PU. El módulo de intra-predicción 126 puede usar varios números de modos de intra-predicción, por ejemplo, 35 modos de intra-predicción direccional. El módulo de intra-predicción 126 puede realizar varias operaciones de intra-predicción. Por ejemplo, el módulo de intra-predicción 126 puede realizar la operación de intra-predicción de ejemplo de la FIG. 6.

[0083] El módulo de predicción 100 puede seleccionar los datos de predicción para una PU entre los datos de predicción generados por el módulo de compensación de movimiento 124 para la PU o los datos de predicción generados por el módulo de intra-predicción 126 para la PU. En algunos ejemplos, el módulo de predicción 100 selecciona los datos de predicción para la PU basándose en las métricas de velocidad / distorsión de los conjuntos de datos de predicción.

[0084] Si el módulo de predicción 100 selecciona datos de predicción generados por el módulo de intra-predicción 126, el módulo de predicción 100 puede señalar el modo de intra-predicción que se utilizó para generar los datos de predicción para la PU, es decir, el modo de intra-predicción seleccionado. El módulo de predicción 100 puede señalar el modo de intra-predicción seleccionado de varias maneras. Por ejemplo, es probable que el modo de intra-predicción seleccionado sea el mismo que el modo de intra-predicción de una PU próxima. En otras palabras, el modo de intra-predicción de la PU próxima puede ser el modo más probable para la PU actual. De este modo, el módulo de predicción 100 puede generar un elemento sintáctico para indicar que el modo de intra-predicción seleccionado es el mismo que el modo de intra-predicción de la PU próxima.

[0085] Después de que el módulo de predicción 100 seleccione los datos de predicción para las PU de una CU, el módulo de generación residual 102 puede generar datos residuales para la CU restando los bloques de vídeo predichos de los datos de predicción de las PU de la CU del bloque de vídeo de la CU. Los datos residuales de una CU pueden incluir bloques de vídeo residuales 2D que corresponden a diferentes componentes de muestra de muestras en el bloque de vídeo de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de vídeo residual que corresponde a diferencias entre componentes de luminancia de muestras en los bloques de vídeo predichos de las PU de la CU y componentes de luminancia de muestras en el bloque de vídeo original de la CU. Además, los datos residuales de la CU pueden incluir bloques de vídeo residuales que corresponden a las diferencias entre componentes de crominancia de muestras en los bloques de vídeo predichos de las PU de la CU y los componentes de crominancia de las muestras en el bloque de vídeo original de la CU .

[0086] El módulo de predicción 100 puede realizar la división en árbol cuádruple para dividir los bloques de vídeo residuales de una CU en sub-bloques. Cada bloque de vídeo residual no dividido puede estar asociado con una TU diferente de la CU. Los tamaños y posiciones de los bloques de vídeo residuales asociados con las TU de una CU pueden o no basarse en los tamaños y posiciones de los bloques de vídeo asociados con las PU de la CU. Una estructura en árbol cuádruple conocida como "árbol cuádruple residual" (RQT) puede incluir nodos asociados con cada uno de los bloques de vídeo residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

[0087] El módulo de transformación 104 puede generar uno o más bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a un bloque de vídeo residual asociado con la TU. Cada uno de los bloques de coeficientes de transformación puede ser una matriz 2D de coeficientes de transformación. El módulo de transformación 104 puede aplicar varias transformaciones al bloque de vídeo residual asociado con una TU. Por ejemplo, el módulo de transformación 104 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de vídeo residual asociado con una TU.

[0088] Después de que el módulo de transformación 104 genere un bloque de coeficientes de transformación asociado con una TU, el módulo de cuantificación 106 puede cuantificar los coeficientes de transformación en el bloque de coeficientes de transformación. El módulo de cuantificación 106 puede cuantificar un bloque de coeficientes de transformación asociado con una TU de una CU basándose en un valor de QP asociado con la CU.

[0089] El codificador de vídeo 20 puede asociar un valor QP con una CU de varias maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de distorsión de velocidad en un bloque de árbol asociado con la CU. En el análisis de distorsión de velocidad, el codificador de vídeo 20 puede generar múltiples representaciones codificadas del bloque de árbol realizando una operación de codificación varias veces en el bloque de árbol. El codificador de vídeo 20 puede asociar diferentes valores de QP con la CU cuando el codificador de vídeo 20 genera diferentes representaciones codificadas del bloque de árbol. El codificador de vídeo 20 puede señalar que un valor QP dado está asociado con la CU cuando el valor QP dado está asociado con la CU en una representación codificada del bloque de árbol que tiene una velocidad de bits y una métrica de distorsión más bajas.

[0090] El módulo de cuantificación inversa 108 y el módulo de transformación inversa 110 pueden aplicar la cuantificación inversa y las transformaciones inversas al bloque de coeficientes de transformación, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformación. El módulo de reconstrucción 112 puede añadir el bloque de vídeo residual reconstruido a las muestras correspondientes de uno o más bloques de vídeo predichos generados por el módulo de predicción 100 para producir un bloque de vídeo reconstruido asociado con una TU. Mediante la reconstrucción de bloques de vídeo para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de vídeo de la CU.

[0091] Después de que el módulo de reconstrucción 112 reconstruya el bloque de vídeo de una CU, el módulo de filtro 113 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en el bloque de vídeo asociado con la CU. Después de realizar las operaciones de desbloqueo, el módulo de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en la memoria intermedia de imágenes descodificadas 114. El módulo de estimación de movimiento 122 y el módulo de compensación de movimiento 124 pueden utilizar una imagen de referencia que contiene el bloque de vídeo reconstruido para realizar inter-predicción en las PU de las imágenes posteriores. Además, el módulo de intra-predicción 126 puede usar bloques de vídeo reconstruidos en la memoria intermedia de imágenes descodificadas 114 para realizar intra-predicción en otras PU en la misma imagen que la

CU.

[0092] El módulo de codificación por entropía 116 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, el módulo de codificación por entropía 116 puede recibir bloques de coeficientes de transformación del módulo de cuantificación 106 y puede recibir elementos sintácticos del módulo de predicción 100. Cuando el módulo de codificación por entropía 116 recibe los datos, el módulo de codificación por entropía 116 puede realizar una o más operaciones de codificación por entropía para generar datos codificados por entropía. Por ejemplo, el codificador de vídeo 20 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación de codificación aritmética binaria adaptable al contexto (CABAC), una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalo de probabilidad (PIPE) u otro tipo de operación de codificación por entropía en los datos. El módulo de codificación por entropía 116 puede enviar un flujo de bits que incluye los datos codificados por entropía.

[0093] Como parte de la realización de una operación de codificación por entropía en los datos, el módulo de codificación por entropía 116 puede seleccionar un modelo de contexto. Si el módulo de codificación por entropía 116 está realizando una operación CABAC, el modelo de contexto puede indicar las probabilidades de que un bin tenga unos valores particulares. En el contexto de CABAC, el término "bin" se utiliza para referirse a un bit de una versión binarizada de un elemento sintáctico.

[0094] Si el módulo de codificación por entropía 116 está realizando una operación CAVLC, el modelo de contexto puede asignar coeficientes a las correspondientes palabras de código. Las palabras de código en CAVLC pueden construirse de forma que los códigos relativamente cortos correspondan a símbolos más probables, mientras que los códigos relativamente largos correspondan a símbolos menos probables. La selección de un modelo de contexto apropiado puede afectar a la eficiencia de codificación de la operación de codificación por entropía.

[0095] En algún ejemplo, el codificador de vídeo 20 puede dividir las CU de un fragmento en una pluralidad de elementos. El codificador de vídeo 20 puede dividir las CU en elementos definiendo dos o más límites horizontales para una imagen y dos o más límites verticales para la imagen. Los límites horizontales pueden incluir los bordes superior e inferior de la imagen. Los límites verticales pueden incluir los lados izquierdo y derecho de la imagen. Cuando el codificador de vídeo no utiliza elementos, el codificador de vídeo 20 puede codificar todos los bloques de árbol de la trama en orden de barrido. Sin embargo, cuando el codificador de vídeo 20 utiliza elementos, el codificador de vídeo puede codificar los elementos en orden de barrido. Cuando el codificador de vídeo 20 codifica un elemento, el codificador de vídeo 20 puede codificar los bloques de árbol (es decir, LCU) dentro del elemento en orden de barrido. De este modo, el codificador de vídeo 20 puede usar elementos para cambiar el orden de codificación de los bloques de árbol de la imagen.

[0096] Además, en algunos casos, cuando el codificador de vídeo 20 está codificando una CU dada, el codificador de vídeo 20 puede utilizar la información asociada con CU espacialmente próximas para llevar a cabo intra-predicción en la CU dada, siempre que la CU dada y las CU espacialmente próximas pertenezcan al mismo elemento. Las CU espacialmente próximas pueden ser CU que están en la misma imagen que la CU dada. (Cuando el codificador de vídeo 20 realiza inter-predicción en la CU dada, el codificador de vídeo 20 también puede usar información de movimiento y de píxeles de las CU temporalmente próximas, independientemente de los elementos de las CU temporalmente próximas). De manera similar, en algunos casos, cuando el codificador de vídeo 20 está codificando una CU dada, el codificador de vídeo 20 puede utilizar información asociada con CU espacialmente próximas para seleccionar un contexto para la codificación por entropía de un elemento sintáctico de la CU dada, siempre y cuando la CU dada y las CU espacialmente próximas estén en el mismo elemento. Debido a estas restricciones, el codificador de vídeo 20 puede ser capaz de codificar dos o más de los elementos en paralelo.

[0097] La FIG. 5 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo 30 que puede implementar las técnicas de esta divulgación. La FIG. 5 se proporciona con fines de explicación y no se limita a las técnicas ampliamente ejemplificadas y descritas en esta divulgación. Con fines de explicación, esta divulgación describe el descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación.

[0098] En el ejemplo de la FIG. 5, el descodificador de vídeo 30 incluye una pluralidad de componentes funcionales. Los componentes funcionales del descodificador de vídeo 30 incluyen un módulo de descodificación por entropía 150, un módulo de predicción 152, un módulo de cuantificación inversa 154, un módulo de transformación inversa 156, un módulo de reconstrucción 158, un módulo de filtro 159 y una memoria intermedia de imágenes descodificadas 160. El módulo de predicción 152 incluye un módulo de compensación de movimiento 162 y un módulo de intra-predicción 164. En algunos ejemplos, el descodificador de vídeo 30 puede realizar una pasada de descodificación en general recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 4. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

[0099] El descodificador de vídeo 30 puede recibir un flujo de bits que comprende datos de vídeo codificados. El

flujo de bits puede incluir una pluralidad de elementos sintácticos. Cuando el descodificador de vídeo 30 recibe el flujo de bits, el módulo de descodificación por entropía 150 puede realizar una operación de análisis sintáctico en el flujo de bits. Como resultado de realizar la operación de análisis sintáctico en el flujo de bits, el módulo de descodificación por entropía 150 puede extraer elementos sintácticos del flujo de bits. Como parte de la realización de la operación de análisis sintáctico, el módulo de descodificación por entropía 150 puede realizar la descodificación por entropía de elementos sintácticos sometidos a codificación por entropía en el flujo de bits. El módulo de predicción 152, el módulo de cuantificación inversa 154, el módulo de transformación inversa 156, el módulo de reconstrucción 158 y el módulo de filtro 159 pueden realizar una operación de reconstrucción que genera datos de vídeo descodificados basados en los elementos sintácticos extraídos del flujo de bits.

[0100] Como se analizó anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de conjunto de parámetros de secuencia, unidades NAL de conjunto de parámetros de imagen, unidades SEI NAL, etc. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, el módulo de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía conjuntos de parámetros de secuencia a partir de unidades NAL de conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen a partir de unidades NAL de conjuntos de parámetros de imagen, datos SEI a partir de unidades SEI NAL, etc.

[0101] Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, el módulo de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía fragmentos codificados a partir de las unidades de NAL de fragmentos codificado. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento. Los elementos sintácticos en la cabecera de fragmento pueden incluir un elemento sintáctico que identifica un conjunto de parámetros de imagen asociado con una imagen que contiene el fragmento. El módulo de descodificación por entropía 150 puede realizar una operación de descodificación por entropía, tal como una operación de descodificación de CAVLC, en la cabecera de fragmento codificada para recuperar la cabecera de fragmento.

[0102] Después de extraer los datos de fragmentos de unidades NAL de fragmentos codificados, el módulo de descodificación por entropía 150 puede extraer bloques de árbol codificados a partir de los datos de fragmento. El módulo de descodificación por entropía 150 puede entonces extraer CU codificadas de los bloques de árbol codificados. El módulo de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen elementos sintácticos de las CU codificadas. Los elementos sintácticos extraídos pueden incluir bloques de coeficientes de transformación codificados por entropía. El módulo de descodificación por entropía 150 puede entonces realizar operaciones de descodificación por entropía en los elementos sintácticos. Por ejemplo, el módulo de descodificación por entropía 150 puede realizar operaciones CABAC en los bloques de coeficientes de transformación.

[0103] Después de que el módulo de descodificación por entropía 150 realice una operación de análisis sintáctico en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en la CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de vídeo residual asociado con la CU.

[0104] Como parte de la realización de una operación de reconstrucción en una TU, el módulo de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, un bloque de coeficientes de transformación asociado con la TU. El módulo de cuantificación inversa 154 puede cuantificar de forma inversa el bloque de coeficientes de transformación de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por el estándar de descodificación H.264. El módulo de cuantificación inversa 154 puede utilizar un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para una CU del bloque de coeficientes de transformación para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa para el módulo de cuantificación inversa 154 a aplicar.

[0105] Después de que el módulo de cuantificación inversa 154 cuantifique de forma inversa un bloque de coeficientes de transformación, el módulo de transformación inversa 156 puede generar un bloque de vídeo residual para la TU asociada con el bloque de coeficientes de transformación. El módulo de transformación inversa 156 puede aplicar una transformación inversa al bloque de coeficientes de transformación para generar el bloque de vídeo residual para la TU. Por ejemplo, el módulo de transformación inversa 156 puede aplicar una DCT inversa, una transformación de número entero inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes de transformación.

[0106] En algunos ejemplos, el módulo de transformación inversa 156 puede determinar una transformación inversa para aplicar al bloque de coeficientes de transformación basándose en la señalización desde el codificador de vídeo 20. En tales ejemplos, el módulo de transformación inversa 156 puede determinar la transformación inversa basada

en una transformación señalada en el nodo raíz de un árbol cuádruple para un bloque de árbol asociado con el bloque de coeficientes de transformación. En otros ejemplos, el módulo de transformación inversa 156 puede inferir la transformación inversa a partir de una o más características de codificación, tales como tamaño de bloque, modo de codificación o similares. En algunos ejemplos, el módulo de transformación inversa 156 puede aplicar una transformación inversa en cascada.

[0107] Si una PU de la CU fue codificada usando inter-predicción, el módulo de compensación de movimiento 162 puede llevar a cabo la compensación de movimiento para generar un bloque de vídeo predicho para la PU. El módulo de compensación de movimiento 162 puede utilizar información de movimiento para que la PU identifique una muestra de referencia para la PU. La muestra de referencia de una PU puede estar en una imagen temporal diferente a la PU. La información de movimiento para la PU puede incluir un vector de movimiento, un índice de imagen de referencia y una dirección de predicción. El módulo de compensación de movimiento 162 puede utilizar la muestra de referencia para que la PU genere el bloque de vídeo predicho para la PU. En algunos ejemplos, el módulo de compensación de movimiento 162 puede predecir la información de movimiento para la PU basándose en la información de movimiento de las PU que están próximas a la PU. En esta divulgación, una PU es una PU inter-predicha si el codificador de vídeo 20 utiliza la inter-predicción para generar el bloque de vídeo predicho de la PU.

[0108] En algunos ejemplos, el módulo de compensación de movimiento 162 puede refinar el bloque de vídeo predicho de una PU mediante la realización de interpolación basándose en filtros de interpolación. Los identificadores para los filtros de interpolación que van a usarse para la compensación de movimiento con una precisión de sub-muestra pueden incluirse en los elementos sintácticos. El módulo de compensación de movimiento 162 puede usar los mismos filtros de interpolación usados por el codificador de vídeo 20 durante la generación del bloque de vídeo predicho de la PU para calcular valores interpolados para muestras de sub-enteros de un bloque de referencia. El módulo de compensación de movimiento 162 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información sintáctica recibida y usar los filtros de interpolación para producir el bloque de vídeo predicho.

[0109] Si se codifica una PU usando intra-predicción, el módulo de intra-predicción 164 puede realizar intra-predicción para generar un bloque de vídeo predicho para la PU. Por ejemplo, el módulo de intra-predicción 164 puede determinar un modo de intra-predicción para la PU basándose en elementos sintácticos en el flujo de bits. El flujo de bits puede incluir elementos sintácticos que el módulo de intra-predicción 164 puede usar para predecir el modo de intra-predicción de la PU. Por ejemplo, el módulo de intra-predicción 164 puede realizar la operación de intra-predicción de ejemplo de la FIG. 6.

[0110] En algunos casos, los elementos sintácticos pueden indicar que el módulo de intra-predicción 164 tiene que utilizar el modo de intra-predicción de otra PU para predecir el modo de intra-predicción de la PU actual. Por ejemplo, puede ser probable que el modo de intra-predicción de la PU actual sea el mismo que el modo de intra-predicción de una PU próxima. En otras palabras, el modo de intra-predicción de la PU próxima puede ser el modo más probable para la PU actual. Por lo tanto, en este ejemplo, el flujo de bits puede incluir un elemento sintáctico pequeño que indica que el modo de intra-predicción de la PU es el mismo que el modo de intra-predicción de la PU próxima. El módulo de intra-predicción 164 puede entonces utilizar el modo de intra-predicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU basándose en los bloques de vídeo de las PU espacialmente próximas.

[0111] El módulo de reconstrucción 158 puede utilizar los bloques de vídeo residuales asociados con las TU de una CU y los bloques de vídeo predichos de las PU de la CU, es decir, datos de intra-predicción o datos de inter-predicción, según corresponda, para reconstruir el bloque de vídeo de la CU. De este modo, el descodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual basándose en los elementos sintácticos en el flujo de bits y puede generar un bloque de vídeo basándose en el bloque de vídeo predicho y el bloque de vídeo residual. Después de que el módulo de reconstrucción 158 reconstruya el bloque de vídeo de la CU, el módulo de filtro 159 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU.

[0112] La memoria intermedia de imágenes descodificadas 160 puede almacenar muestras descodificadas para imágenes de los datos de vídeo. Así, después de que el módulo de filtro 159 realice la operación de desbloqueo, el descodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en la memoria intermedia de imágenes descodificadas 160. La memoria intermedia de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intra-predicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de vídeo en la memoria intermedia de imágenes descodificadas 160, operaciones de intra-predicción o de inter-predicción en las PU de otras CU.

[0113] La FIG. 6 es un diagrama de flujo que ilustra un ejemplo de operación de intra-predicción 200. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede realizar la operación de intra-predicción 200. En otros ejemplos, el codificador de vídeo puede usar operaciones de intra-predicción distintas de la operación de intra-predicción 200. Por ejemplo, en otros ejemplos, el codificador de vídeo puede realizar una

operación de intra-predicción en la que el codificador de vídeo realiza más, menos o diferentes pasos que la operación de intra-predicción 200.

5 **[0114]** Después de que el codificador de vídeo inicie la operación de intra-predicción 200, el codificador de vídeo puede realizar una operación de relleno para una PU (202). Cuando el codificador de vídeo realiza la operación de relleno, el codificador de vídeo puede asignar valores a píxeles de borde no disponibles asociados con la PU. El codificador de vídeo puede realizar varias operaciones de relleno. Por ejemplo, el codificador de vídeo puede realizar la operación de relleno de ejemplo de la FIG. 7, la operación de relleno de ejemplo de la FIG. 8, la operación de relleno de ejemplo de la FIG. 10, u otra operación de relleno.

10 **[0115]** Después de que el codificador de vídeo realice la operación de relleno, el codificador de vídeo puede generar, basándose en los píxeles de borde, un bloque de vídeo intra-predicho para la PU (204). El codificador de vídeo puede usar algunos o todos los píxeles de borde para generar el bloque de vídeo intra-predicho.

15 **[0116]** La FIG. 7 es un diagrama de flujo que ilustra una operación de relleno 250 de ejemplo. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede realizar la operación de relleno 250. En otros ejemplos, el codificador de vídeo puede utilizar operaciones de relleno distintas de la operación de relleno 250. Por ejemplo, en otros ejemplos, el codificador de vídeo puede realizar una operación de relleno en la que el codificador de vídeo realiza más, menos o diferentes pasos que la operación de relleno 250.

20 **[0117]** Después de que el codificador de vídeo inicie la operación de relleno 250, el codificador de vídeo puede establecer un indicador de píxel para indicar un píxel de borde inferior izquierdo (252). Por ejemplo, en el ejemplo de la FIG. 2, el codificador de vídeo puede establecer el indicador de píxel para indicar el píxel de borde P. Para facilitar la explicación, esta divulgación puede referirse al píxel de borde indicado por el indicador de píxel como el píxel de borde actual.

25 **[0118]** A continuación, el codificador de vídeo puede determinar si el píxel de borde actual está disponible (254). El codificador de vídeo puede determinar si el píxel de borde actual está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si el píxel de borde actual está disponible.

30 **[0119]** En respuesta a determinar que el píxel de borde actual no está disponible ("NO" de 254), el codificador de vídeo puede asignar un valor predeterminado al píxel de borde actual (256). Debido a que el píxel de borde actual es el píxel de borde inferior izquierdo, el codificador de vídeo puede determinar si el píxel de borde inferior izquierdo está disponible y asignar un valor predeterminado al píxel de borde inferior izquierdo en respuesta a determinar que el píxel de borde inferior izquierdo no está disponible. En otro ejemplo, el codificador de vídeo puede asignar un valor de un siguiente píxel de borde disponible al píxel de borde inferior izquierdo en respuesta a determinar que el píxel de borde inferior izquierdo no está disponible.

35 **[0120]** Después de asignar un valor a la píxel de borde actual o en respuesta a determinar que el píxel de borde actual está disponible ("SÍ" de 254), el codificador de vídeo puede actualizar el indicador de píxel para indicar un siguiente píxel de borde (258). Si el píxel de borde actual está entre el píxel de borde inferior izquierdo y el píxel de borde de esquina (por ejemplo, el píxel de borde R en la FIG. 2), el siguiente píxel de borde está inmediatamente por encima del píxel de borde actual. Por ejemplo, en el ejemplo de la FIG. 2, si el píxel de borde actual es el píxel de borde P, el siguiente píxel de borde es el píxel de borde O. Si el píxel de borde actual es el píxel de borde O, el siguiente píxel de borde es el píxel de borde N, etc. Si el píxel actual es el píxel de borde de esquina o entre el píxel de borde de esquina y el píxel de borde superior derecho (por ejemplo, el píxel de borde H en la FIG. 2), el siguiente píxel de borde está inmediatamente a la derecha del píxel de borde actual. Por ejemplo, en el ejemplo de la FIG. 2, si el píxel de borde actual es el píxel de borde R, el siguiente píxel de borde es el píxel de borde A. Si el píxel de borde actual es el píxel de borde A, el siguiente píxel de borde es el píxel de borde B, etc. Al actualizar el indicador de píxel, el siguiente píxel de borde se convierte en el píxel de borde actual.

40 **[0121]** Después de actualizar el indicador de píxel, el codificador de vídeo puede determinar si el píxel de borde actual está disponible (260). En respuesta a la determinación de que no está disponible el píxel de borde actual ("NO" de 260), el codificador de vídeo puede asignar el valor del píxel de borde anterior al píxel de borde actual (262). En otras palabras, el codificador de vídeo puede asignar al píxel de borde actual un valor de un píxel de borde que está inmediatamente antes del píxel de borde actual de acuerdo con el orden. Por ejemplo, en el ejemplo de la FIG. 2, si el píxel de borde actual es el píxel de borde M y el píxel de borde M no está disponible, el codificador de vídeo puede asignar el valor de píxel de borde N al píxel de borde M. Si el píxel de borde actual es el píxel de borde L y el píxel de borde L no está disponible, el codificador de vídeo puede asignar el valor de píxel de borde M al píxel de borde L, etc. Cuando el codificador de vídeo asigna un valor de un primer píxel de borde a un segundo píxel de borde, el codificador de vídeo tal vez no cambie el valor real del segundo píxel de borde, pero puede usar el valor del primer píxel de borde como el valor del segundo píxel de borde con el fin de generar un bloque de vídeo intra-predicho.

45 **[0122]** Después de realizar el paso 262 o en respuesta a determinar que el píxel de borde actual está disponible

("Sí" de 260), el codificador de vídeo puede determinar si el píxel de borde actual es el píxel de borde superior derecho (264). Por ejemplo, en el ejemplo de la FIG. 2, el codificador de vídeo puede determinar si el píxel de borde actual es el píxel de borde H. El codificador de vídeo puede finalizar la operación de relleno 250 en respuesta a determinar que el píxel de borde actual es el píxel de borde superior derecho ("Sí" de 264). Por otra parte, en respuesta a la determinación de que el píxel de borde actual no es el píxel de borde superior derecho ("NO" de 264), el codificador de vídeo puede actualizar de nuevo el indicador de píxel (258). El codificador de vídeo puede entonces repetir los pasos 260-264 con respecto al nuevo píxel de borde actual.

[0123] De esta manera, el codificador de vídeo puede realizar una operación de relleno que procesa un conjunto de píxeles de borde de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y procede a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho. Cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel previamente procesado por la operación de relleno.

[0124] En un ejemplo de operación de relleno similar a la operación de relleno 250, (x, y) indica las coordenadas del bloque de vídeo y (x-1, y-1) indica las coordenadas del píxel de borde de esquina. Si (x-1, y-1) no está disponible, el codificador de vídeo puede comprobar el predictor superior de izquierda a derecha hasta que el codificador de vídeo alcance un píxel de borde disponible. Si el píxel de borde disponible tiene coordenadas (x+d, y-1), el codificador de vídeo puede asignar el valor del píxel de borde en (x+d, y-1) a todos los píxeles de borde de (x-1, y-1) a (x+d-1, y-1). De lo contrario, si el píxel de borde en (x-1, y-1) está disponible, el codificador de vídeo puede establecer una variable d en -1 y ejecutar el siguiente pseudo-código:

```

para (i = d; i < 2N; i++) {
    si el píxel de borde en (x+i, y-1) está disponible, no haga nada
    si el píxel de borde en (x+i, y-1) no está disponible,
        asigne el valor del píxel de borde en (x+i-1, y-1) al píxel de borde
        en (x+i, y-1)
}

```

En el pseudo-código anterior, N puede indicar la anchura del bloque de vídeo. Además, el codificador de vídeo puede determinar si el píxel de borde en (x-1, y) está disponible. Si el píxel de borde en (x-1, y) no está disponible, el codificador de vídeo puede comprobar el predictor izquierdo de arriba a abajo hasta que el codificador de vídeo alcance un píxel de borde disponible. Si el píxel de borde disponible tiene coordenadas (x-1, y+d), el codificador de vídeo puede asignar el valor del píxel de borde en (x-1, y+d) a todos los píxeles de borde de (x-1, y) a (x-1, y+d-1). De lo contrario, si el píxel de borde en (x-1, y) está disponible, el codificador de vídeo puede establecer una variable d en 0 y ejecutar el siguiente pseudo-código:

```

para (i = d; i < 2N; i++) {
    si el píxel de borde en (x-1, y+i) está disponible, no haga nada
    si el píxel de borde en (x-1, y+i) no está disponible,
        asigne el valor del píxel de borde en (x-1, y+i-1) al píxel de borde
        en (x-1, y+i)
}

```

[0125] En el ejemplo de la FIG. 7, para $x = -1$, $y = nS*2-2 \dots - 1$, el codificador de vídeo puede sustituir un valor de $p[x, y+1]$ por un valor de $p[x, y]$ si $p[x, y]$ es un píxel de borde no disponible. Para $x = 0$, $y = nS*2-1, y-1$, sustituyendo un valor de $p[x-1, y]$ por un valor de $p[x, y]$ si $p[x, y]$ es un píxel de borde no disponible. nS es igual a una anchura o altura del bloque de vídeo intra-predicho, p es una matriz bidimensional de muestras y $p[0,0]$ es una muestra de luma superior izquierda del bloque de vídeo intra-predicho.

[0126] La FIG. 8 es un diagrama conceptual que ilustra un orden en el que el ejemplo de operación de relleno de la FIG. 7 procede a través de píxeles de borde. Como se ilustra en el ejemplo de la FIG. 8, la operación de relleno comienza en el píxel de borde inferior izquierdo, procede hacia arriba hasta el píxel de borde de esquina y, a continuación, procede hacia la derecha hasta el píxel de borde superior derecho.

[0127] La FIG. 9 es un diagrama de flujo que ilustra una operación de relleno 300 de ejemplo. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede realizar la operación de relleno 300. En otros ejemplos, el codificador de vídeo puede utilizar operaciones de relleno distintas de la operación de relleno 300. Por ejemplo, en otros ejemplos, el codificador de vídeo puede realizar una operación de relleno en la que el codificador de vídeo realiza más, menos o diferentes pasos que la operación de relleno 300.

[0128] Después de que el codificador de vídeo inicia la operación de relleno 300, el codificador de vídeo puede establecer un primer indicador de píxel y un segundo indicador de píxel para indicar un punto de división (302). El punto de división puede dividir los píxeles de borde en dos segmentos. Uno de los segmentos puede incluir píxeles de borde que están por encima y a la derecha del punto de división. El otro de los segmentos puede incluir píxeles de borde que están por debajo y a la izquierda del punto de división.

[0129] El punto de división pueden ser varios de los píxeles de borde. Por ejemplo, el codificador de vídeo puede establecer el primer y el segundo indicadores de píxel en el píxel de borde de esquina (por ejemplo, el píxel de borde R en la FIG. 2). En otro ejemplo, el codificador de vídeo puede establecer el primer indicador de píxel y el segundo indicador de píxel para indicar otro píxel de borde (por ejemplo, el píxel de borde L en la FIG. 2).

[0130] Después de establecer el primero y el segundo indicadores de píxel para indicar el punto de división, el codificador de vídeo puede determinar si el punto de división está disponible (304). El codificador de vídeo puede determinar si el punto de división está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si el punto de división está disponible.

[0131] En respuesta a determinar que el punto de división no está disponible ("NO" de 304), el codificador de vídeo puede obtener un valor del punto de división (306). El codificador de vídeo puede obtener el valor del punto de división de varias maneras. Por ejemplo, el codificador de vídeo puede determinar si está disponible cualquiera de los píxeles de borde adyacentes al punto de división. En este ejemplo, si uno de los píxeles de borde adyacentes al punto de división está disponible, el codificador de vídeo puede obtener el valor del punto de división asignando el valor del píxel de borde adyacente al punto de división. Además, en este ejemplo, el codificador de vídeo puede asignar un valor predeterminado al punto de división si ninguno de los píxeles de borde adyacentes está disponible. Por ejemplo, si el punto de división es el punto de división de esquina, el codificador de vídeo puede utilizar el siguiente pseudo-código para obtener un valor del punto de división:

Si P_{L1} está disponible,

$$P_{\text{pad}} = P_{L1},$$

en caso contrario, si P_{A1} está disponible,

$$P_{\text{pad}} = P_{A1},$$

en caso contrario,

$P_{\text{pad}} =$ un valor predeterminado.

En el pseudo-código anterior, P_{L1} puede indicar el valor del píxel de borde por debajo del píxel de borde de esquina, P_{pad} puede indicar el valor obtenido que se ha de asignar al píxel de borde de esquina y P_{A1} puede indicar el valor del píxel de borde a la derecha del píxel de borde de esquina. En el caso de que el píxel de borde de esquina y el píxel de borde por debajo y el píxel de borde a la derecha del píxel de borde de esquina no estén disponibles, el uso de un valor predeterminado para el píxel de borde de esquina puede tener poco impacto en el rendimiento de codificación, ya que este caso puede producirse raramente.

[0132] Después de obtener el valor del punto de división o después de determinar que el punto de división está disponible ("SÍ" de 304), el codificador de vídeo puede actualizar el primer indicador de píxel para indicar un píxel de borde que está por debajo o a la izquierda del píxel de borde indicado por el primer indicador de píxel (308). Para facilitar la explicación, esta divulgación puede referirse al píxel de borde indicado por el primer indicador de píxel como el primer píxel de borde actual. Por ejemplo, en el ejemplo de la FIG. 2, si el primer píxel de borde actual es el píxel de borde B, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar el píxel de borde A. Si el primer píxel de borde actual es el píxel de borde R, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar el píxel de borde I. Si el primer píxel de borde actual es el píxel de borde I, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar el píxel de borde J.

[0133] Después de actualizar el primer indicador de píxel, el codificador de vídeo puede determinar si el primer píxel de borde actual está disponible (310). El codificador de vídeo puede determinar si el primer píxel de borde actual está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si el primer píxel de borde actual está disponible.

[0134] En respuesta a determinar que el primer píxel de borde actual no está disponible ("NO" de 310), el codificador de vídeo puede asignar el valor del píxel de borde anterior al primer píxel de borde actual (312). Por ejemplo, en el ejemplo de la FIG. 2, si el primer píxel de borde actual es el píxel de borde I y el píxel de borde I no está disponible, el codificador de vídeo puede asignar el valor del píxel de borde R al píxel de borde I. Si el primer píxel de borde actual es el píxel de borde J y el píxel de borde J no está disponible, el codificador de vídeo puede asignar el valor del píxel de borde I al borde del píxel J, etc.

[0135] Una vez que el codificador de vídeo ha realizado la etapa 310 o después de determinar que el primer píxel de borde actual está disponible ("SÍ" de 310), el codificador de vídeo puede determinar si el primer píxel de borde actual es el píxel de borde inferior izquierdo (314). En respuesta a la determinación de que el primer píxel de borde actual es el píxel de borde inferior izquierdo ("SÍ" de 314), el codificador de vídeo puede finalizar la operación de relleno

300 con respecto a píxeles de borde que ocurren a la izquierda o debajo del píxel de borde de punto de división. Sin embargo, en respuesta a determinar que el primer píxel de borde actual no es el píxel de borde inferior izquierdo ("NO" de 314), el codificador de vídeo puede actualizar de nuevo el primer indicador de píxel (308) y llevar a cabo los pasos 310-314 con respecto a otro píxel de borde. De esta manera, la operación de relleno 300 puede asignar valores a cada píxel de borde no disponible a la izquierda o por debajo del píxel de borde de punto de división.

[0136] Además, después de obtener el valor del píxel de borde punto de división en el paso 306 o en respuesta a la determinación de que el punto de división está disponible ("SÍ" de 304), el codificador de vídeo puede actualizar el segundo indicador de píxel para indicar un píxel de borde que está por encima o a la derecha del píxel de borde indicado por el segundo indicador de píxel (316). Para facilitar la explicación, esta divulgación puede referirse al píxel de borde indicado por el segundo indicador de píxel como el segundo píxel de borde actual.

[0137] Después de actualizar el segundo indicador de píxel, el codificador de vídeo puede determinar si el segundo píxel de borde actual está disponible (318). El codificador de vídeo puede determinar si el píxel de borde actual está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si está disponible el segundo píxel de borde actual.

[0138] En respuesta a determinar que el segundo píxel de borde actual no está disponible ("NO" de 318), el codificador de vídeo puede asignar el valor de un píxel de borde anterior al segundo píxel de borde actual (320). Por ejemplo, en el ejemplo de la FIG. 2, si el segundo píxel de borde actual es el píxel de borde D y el píxel de borde D no está disponible, el codificador de vídeo puede asignar el valor del píxel de borde C al píxel de borde D.

[0139] Después de realizar la etapa 320 o después de que el codificador de vídeo haya determinado que el segundo píxel de borde actual está disponible ("SÍ" de 318), el codificador de vídeo puede determinar si el segundo píxel de borde actual es el píxel de borde superior derecho (322). En respuesta a determinar que el segundo píxel de borde actual es el píxel de borde superior derecho, el codificador de vídeo puede finalizar la operación de relleno 300 con respecto a píxeles de borde que están a la derecha o por encima del píxel de borde de punto de división ("SÍ" de 322). Sin embargo, en respuesta a determinar que el segundo píxel de borde actual no es el píxel de borde superior derecho ("NO" de 322), el codificador de vídeo puede actualizar el segundo indicador de píxel (316) y realizar los pasos 318-322 con respecto a otro píxel de borde. De esta manera, el codificador de vídeo puede procesar cada píxel de borde que está por encima o a la derecha del punto de división.

[0140] En el ejemplo de la FIG. 9, el codificador de vídeo puede realizar los pasos 308-314 y los pasos 316-322 en paralelo. En otros ejemplos, el codificador de vídeo puede realizar los pasos 308-314 y los pasos 316-322 secuencialmente.

[0141] La FIG. 10 es un diagrama conceptual que ilustra un ejemplo de orden en el que la operación de relleno de la FIG. 9 procede a través de píxeles de borde. Como se ilustra en el ejemplo de la FIG. 10, el píxel de borde de esquina es el píxel de borde de punto de división. El codificador de vídeo puede procesar simultáneamente píxeles de borde por debajo del píxel de borde de esquina (por ejemplo, en los pasos 308-314) y píxeles de borde a la derecha del píxel de borde de esquina (por ejemplo, pasos 316-322). En el ejemplo de la FIG. 10, el píxel de borde de esquina está sombreado. Si el píxel de borde de esquina no está disponible, el codificador de vídeo puede obtener un valor para el píxel de borde de esquina desde uno o ambos píxeles de borde próximos 340.

[0142] En otro ejemplo, el codificador de vídeo puede comprobar de izquierda a derecha a través del predictor superior hasta que el codificador de vídeo identifica un píxel de borde disponible. El codificador de vídeo puede entonces establecer el píxel de esquina y todos los píxeles de borde entre el píxel de esquina y el píxel de borde disponible en el valor del píxel de borde disponible. En este ejemplo, si el píxel de borde situado por debajo del píxel de esquina no está disponible, el codificador de vídeo puede comprobar de arriba a abajo a lo largo del predictor izquierdo hasta que el codificador de vídeo identifique un píxel de borde disponible. El codificador de vídeo puede entonces establecer cada píxel de borde por debajo del píxel de esquina en el valor del píxel de borde disponible.

[0143] La FIG. 11 es un diagrama de flujo que ilustra una operación de relleno 350 de ejemplo. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede realizar la operación de relleno 350. En otros ejemplos, el codificador de vídeo puede utilizar operaciones de relleno distintas de la operación de relleno 350. Por ejemplo, en otros ejemplos, el codificador de vídeo puede realizar una operación de relleno en la que el codificador de vídeo realiza más, menos o diferentes pasos que la operación de relleno 350.

[0144] En la operación de relleno 350, los píxeles de borde incluyen una pluralidad de puntos de división candidatos. La pluralidad de puntos de división candidatos puede incluir varios de los píxeles de borde. Por ejemplo, la pluralidad de puntos de división candidatos puede incluir el píxel de borde de esquina, un píxel de borde a mitad de camino entre el píxel de borde de esquina y el píxel de borde inferior izquierdo y un píxel de borde a mitad de camino entre el píxel de borde de esquina y el píxel de borde superior derecho. En otro ejemplo, la pluralidad de puntos de división candidatos puede incluir el píxel de borde inferior izquierdo y el píxel de borde superior derecho. En otro ejemplo, la pluralidad de puntos de división candidatos puede incluir el píxel de borde inmediatamente a la derecha del píxel de borde de esquina y el píxel de borde inmediatamente debajo del píxel de borde de esquina.

5 **[0145]** En otro ejemplo, la pluralidad de puntos de división candidatos puede incluir todos los píxeles de borde entre dos píxeles de borde, P_{LX} y P_{AY} . En este ejemplo, P_{LX} puede ser cualquier píxel de borde desde el píxel de borde de esquina hasta un píxel de borde inferior izquierdo y P_{AY} puede ser cualquier píxel de borde desde el píxel de borde de esquina hasta el píxel de borde superior derecho. En este ejemplo, el codificador de vídeo puede buscar a través de los puntos de división candidatos secuencialmente de P_{LX} a P_{AY} o de P_{AY} a P_{LX} un punto de división candidato disponible. La búsqueda de un punto de división candidato disponible de esta manera puede reducir la complejidad de búsqueda a una búsqueda unidimensional. Por ejemplo, en este ejemplo, si el punto de división es el píxel de borde inferior izquierdo, el codificador de vídeo realiza una operación de relleno que procesa el conjunto de píxeles de borde de acuerdo con un orden que comienza en el píxel de borde inferior izquierdo y procede a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho, en el que cuando la operación de relleno procesa un píxel de borde no disponible, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado por la operación de relleno.

15 **[0146]** En otro ejemplo, la pluralidad de puntos de división candidatos puede incluir todos los píxeles de borde. En este ejemplo, el codificador de vídeo puede buscar un punto de división candidato disponible a través de los puntos de división candidatos secuencialmente desde un píxel de borde inferior izquierdo hasta un píxel de borde superior derecho o viceversa. La búsqueda de un punto de división candidato disponible de esta manera puede reducir la complejidad de búsqueda a una búsqueda unidireccional.

20 **[0147]** Después de que el codificador de vídeo inicie la operación de relleno 350, el codificador de vídeo puede establecer un indicador de punto de división para indicar un primero de los puntos de división candidatos (352). Para facilitar la explicación, esta divulgación puede referirse al punto de división candidato indicado por el indicador de punto de división como el punto de división actual. Por ejemplo, si la pluralidad de puntos de división candidatos incluye el píxel de borde de esquina, el codificador de vídeo puede establecer el indicador de punto de división en el píxel de borde de esquina.

30 **[0148]** Después de establecer el indicador de punto de división para indicar uno de los puntos de división candidatos, el codificador de vídeo puede determinar si el punto de división actual está disponible (354). El codificador de vídeo puede determinar si el punto de división actual está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si el punto de división actual está disponible.

35 **[0149]** Si el punto de división actual no está disponible ("NO" de 354), el codificador de vídeo puede determinar si el punto de división actual es el último uno de los puntos de división candidatos a comprobar (356). En respuesta a la determinación de que el punto de división actual es el último de los puntos de división candidatos, el codificador de vídeo puede asignar un valor predeterminado al punto de división actual (358). En otros ejemplos, el codificador de vídeo puede asignar un valor basado en los valores de píxeles próximos al punto de división actual en lugar de usar el valor predeterminado.

40 **[0150]** Por otro lado, en respuesta a determinar que el punto de división actual no es el último de los puntos de división candidatos, el codificador de vídeo puede actualizar el indicador del punto de división de tal manera que el indicador del punto de división indica el siguiente de los puntos de división candidatos (360). Por ejemplo, en el ejemplo de la FIG. 2, la pluralidad de puntos de división candidatos puede incluir el píxel de borde L, el píxel de borde R y el píxel de borde D. Si el punto de división actual es el píxel de borde L y el píxel de borde L no está disponible, el codificador de vídeo puede actualizar el indicador de punto de división tal que el indicador de punto de división indique el píxel de borde R. El codificador de vídeo puede buscar un punto de división candidato disponible de varias maneras. Por ejemplo, el codificador de vídeo puede comenzar con un punto de división candidato que esté más cercano al píxel de borde inferior izquierdo y luego buscar a través de los puntos de división candidatos hacia el punto de división candidato que esté más próximo al píxel de borde superior derecho.

50 **[0151]** Después de actualizar el indicador de punto de división, el codificador de vídeo puede determinar de nuevo si el punto de división actual está disponible (354). De esta manera, el codificador de vídeo puede procesar puntos de división candidatos hasta que el codificador de vídeo determine que uno de los puntos de división candidatos está disponible o hasta que el codificador de vídeo determine que ninguno de los puntos de división candidatos está disponible, o viceversa.

60 **[0152]** Después de realizar el paso 358 o después de determinar que el punto de división actual está disponible ("SÍ" de 354), el codificador de vídeo puede establecer un primer indicador de píxel y un segundo indicador de píxel para indicar el punto de división actual (362). Por ejemplo, si el punto de división actual es el píxel de borde de esquina, el codificador de vídeo puede establecer el primer y el segundo indicadores de píxel para indicar el píxel de borde de esquina (por ejemplo, el píxel de borde R en la FIG. 2).

65 **[0153]** Después de establecer el primer y el segundo indicadores de píxel para indicar el punto de división actual, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar un píxel de borde que está por debajo o a la izquierda del píxel de borde indicado por el primer indicador de píxel (364). Para facilitar la explicación, esta

divulgación puede referirse al píxel de borde indicado por el primer indicador de píxel como el primer píxel de borde actual. Por ejemplo, en el ejemplo de la FIG. 2, si el primer píxel de borde actual es el píxel de borde B, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar el píxel de borde A. Si el primer píxel de borde actual es el píxel de borde R, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar el píxel de borde I. Si el primer píxel de borde actual es el píxel de borde I, el codificador de vídeo puede actualizar el primer indicador de píxel para indicar el píxel de borde J.

[0154] Después de actualizar el primer indicador de píxel, el codificador de vídeo puede determinar si el primer píxel de borde actual está disponible (366). El codificador de vídeo puede determinar si el primer píxel de borde actual está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si el primer píxel de borde actual está disponible.

[0155] En respuesta a determinar que el primer píxel de borde actual no está disponible ("NO" de 366), el codificador de vídeo puede asignar el valor del píxel de borde anterior al primer píxel de borde actual (368). Por ejemplo, en el ejemplo de la FIG. 2, si el primer píxel de borde actual es el píxel de borde I y el píxel de borde I no está disponible, el codificador de vídeo puede asignar el valor de píxel de borde R al píxel de borde I. Si el primer píxel de borde actual es el píxel de borde J y el píxel de borde J no está disponible, el codificador de vídeo puede asignar el valor del píxel de borde I al píxel de borde J, etc.

[0156] Una vez que el codificador de vídeo ha asignado el valor del píxel de borde anterior al primer píxel de borde actual o después de determinar que el primer píxel de borde actual está disponible ("SÍ" de 366), el codificador de vídeo puede determinar si el primer píxel de borde actual es el píxel de borde inferior izquierdo (370). En respuesta a la determinación de que el primer píxel de borde actual es el píxel de borde inferior izquierdo ("SÍ" de 370), el codificador de vídeo puede finalizar la operación de relleno 300 con respecto a píxeles de borde que aparecen a la izquierda o debajo del punto de división actual. Sin embargo, en respuesta a determinar que el primer píxel de borde actual no es el píxel de borde inferior izquierdo ("NO" de 370), el codificador de vídeo puede actualizar de nuevo el primer indicador de píxel (364) y llevar a cabo los pasos 366-370 con respecto a otro píxel de borde. De esta manera, la operación de relleno 350 puede asignar valores a cada píxel de borde no disponible a la izquierda o por debajo del punto de división actual.

[0157] Además, después de establecer el primer y el segundo indicadores de píxel en el punto de división actual en el paso 362, el codificador de vídeo puede actualizar el segundo indicador de píxel para indicar un píxel de borde que esté por encima o a la derecha del píxel de borde indicado por el segundo indicador de píxel (372). Para facilitar la explicación, esta divulgación puede referirse al píxel de borde indicado por el segundo indicador de píxel como el segundo píxel de borde actual.

[0158] Después de actualizar el segundo indicador de píxel, el codificador de vídeo puede determinar si el segundo píxel de borde actual está disponible (374). El codificador de vídeo puede determinar si el segundo píxel de borde actual está disponible de varias maneras. Por ejemplo, el codificador de vídeo puede realizar la operación de ejemplo de la FIG. 13 para determinar si está disponible el segundo píxel de borde actual.

[0159] En respuesta a determinar que el segundo píxel de borde actual no está disponible ("NO" de 374), el codificador de vídeo puede asignar el valor de un píxel de borde anterior al segundo píxel de borde actual (376). Por ejemplo, en el ejemplo de la FIG. 2, si el segundo píxel de borde actual es el píxel de borde D y el píxel de borde D no está disponible, el codificador de vídeo puede asignar el valor del píxel de borde C al píxel de borde D.

[0160] Después de asignar el valor de un píxel de borde anterior al segundo píxel de borde actual o después de que el codificador de vídeo haya determinado que el segundo píxel de borde actual está disponible ("SÍ" de 374), el codificador de vídeo puede determinar si el segundo píxel de borde actual es el píxel de borde superior derecho (378). En respuesta a la determinación de que el segundo píxel de borde actual es el píxel de borde superior derecho, el codificador de vídeo puede finalizar la operación de relleno 300 con respecto a píxeles de borde que están a la derecha o por encima del punto de división actual. Sin embargo, en respuesta a determinar que el segundo píxel de borde actual no es el píxel de borde superior derecho ("NO" de 378), el codificador de vídeo puede actualizar el segundo indicador de píxel (372) y realizar los pasos 374-378 con respecto a otro píxel de borde. De esta manera, el codificador de vídeo puede procesar cada píxel de borde que está por encima o a la derecha del píxel de borde de punto de división.

[0161] En el ejemplo de la FIG. 11, el codificador de vídeo puede realizar los pasos 364-370 y los pasos 372-378 en paralelo. En otros ejemplos, el codificador de vídeo puede realizar los pasos 364-370 y los pasos 372-378 secuencialmente.

[0162] La FIG. 12 es un diagrama conceptual que muestra un ejemplo de orden en el que la operación de relleno de la FIG. 11 procede a través de píxeles de borde. Como se ilustra en el ejemplo de la FIG. 12, los píxeles de borde sombreados son puntos de división candidatos. En el ejemplo de la FIG. 12, el píxel de borde 400 puede ser el punto de división seleccionado y el codificador de vídeo puede procesar simultáneamente píxeles de borde por debajo del punto de división seleccionado (por ejemplo, en los pasos 364-370 de la FIG. 11) y píxeles de borde por encima y a

la derecha del punto de división seleccionado (por ejemplo, los pasos 372-378).

[0163] La FIG. 13 es un diagrama de flujo que ilustra una operación de ejemplo 450 para determinar si un píxel de borde está disponible. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede realizar la operación 450. En otros ejemplos, el codificador de vídeo puede utilizar operaciones distintas de la operación 450 para determinar si está disponible un píxel de borde. Por ejemplo, en otros ejemplos, el codificador de vídeo puede realizar una operación en la que el codificador de vídeo realiza más, menos o diferentes pasos que la operación 450.

[0164] Después de que un codificador de vídeo inicie la operación 450, el codificador de vídeo puede determinar si el píxel de borde está fuera de un borde de una imagen actual (452). La imagen actual puede ser la imagen que el codificador de vídeo está codificando actualmente. Por ejemplo, si el codificador de vídeo está realizando una operación de intra-predicción para una PU cuyo bloque de vídeo es adyacente a un borde superior o izquierdo de la imagen actual, uno o más píxeles por encima o a la izquierda del bloque de vídeo pueden estar fuera del borde de la imagen actual. En respuesta a la determinación de que el píxel de borde está fuera del borde de la imagen actual ("Sí" de 452), el codificador de vídeo puede determinar que el píxel de borde no está disponible (454).

[0165] En respuesta a determinar que el píxel de borde no está fuera del borde de la imagen actual ("NO" de 452), el codificador de vídeo puede determinar si el píxel de borde está fuera de un borde de un fragmento o elemento actual y la predicción de fragmento cruzado / elemento cruzado no está permitida (456). El fragmento o elemento actual puede ser el fragmento o el elemento que el codificador de vídeo está codificando actualmente. Por ejemplo, si el codificador de vídeo está realizando una operación de intra-predicción para una PU cuyo bloque de vídeo es adyacente a un borde superior o izquierdo del fragmento o elemento actual, puede haber uno o más píxeles por encima o a la izquierda del bloque de vídeo fuera del borde del fragmento o elemento actual. Si está permitida la predicción de fragmentos cruzados, está permitido que el codificador de vídeo utilice valores de píxeles en un primer fragmento para generar un bloque de vídeo intra-predicho para una CU en un segundo fragmento. En respuesta a la determinación de que el píxel de borde está fuera del borde del fragmento o elemento actual y que la predicción de fragmento / elemento cruzado no está permitida ("Sí" de 456), el codificador de vídeo puede determinar que el píxel de borde no está disponible (454).

[0166] Sin embargo, en respuesta a determinar que el píxel de borde no está fuera del fragmento o elemento actual o que la predicción de fragmento cruzado / elemento cruzado está permitida ("NO" de 456), el codificador de vídeo puede determinar si el píxel de borde está asociado con una CU inter-predicha y si la intra-predicción restringida está habilitada (458). En otras palabras, si el píxel de borde pertenece a una CU inter-predicha y se habilita la intra-predicción restringida, el píxel de borde puede marcarse como no disponible o sin disponibilidad. El píxel de borde puede estar asociado con una CU inter-predicha si el píxel de borde está en un bloque de vídeo asociado con una CU inter-predicha.

[0167] En H.264/AVC, un macrobloque (MB) en un fragmento inter-predicho puede estar codificado como MB inter o MB intra. De manera similar, en HEVC, una CU en un fragmento P o un fragmento B puede ser una CU inter-predicha o una CU intra-predicha. Los MB intra y las CU intra-predichas se predicen a partir de los bloques de vídeo próximos. Además, un codificador de vídeo puede codificar MB intra restringidas. Un codificador de vídeo puede generar un MB intra restringido a partir de MB intra próximos en el mismo fragmento. De acuerdo con las técnicas de esta divulgación, un codificador de vídeo puede generar una CU restringida intra-predicha a partir de los bloques de vídeo de CU próximas intra-predichas, pero no bloques de vídeo de CU próximas inter-predichas. Un `constrained_intra_pred_flag` puede señalarse en un conjunto de parámetros de imagen (PPS) para especificar si un grupo de imágenes se codifica con los únicos modos intra restringidos e inter.

[0168] La actualización intra puede ser una herramienta de resistencia a los errores que permite a un codificador de vídeo insertar MB intra o CU intra-predichas con el fin de reducir la propagación de errores. En el caso de que se pierda una imagen de referencia, un descodificador de vídeo tal vez no sea capaz de descodificar correctamente MB inter o CU inter-predichas que usen la imagen de referencia. Si los MB inter o las CU inter-predichas se utilizan para predecir un MB intra o una CU intra-predicha, el codificador de vídeo puede ser incapaz de descodificar el MB intra o la CU intra-predicha. Por lo tanto, el error puede propagarse al MB intra o la CU intra-predicha. Debido a que las MB intra restringidas y las CU intra-predichas restringidas no se basan en MB inter o CU inter-predichas, la pérdida de la imagen de referencia no impide que un descodificador de vídeo descodifique los MB intra restringidos o las CU intra-predichas. Por lo tanto, puede ser más resistente a los errores codificar los MB intra o las CU intra-predichas como MB intra restringidos o CU intra-predichas restringidas.

[0169] En la codificación de vídeo escalable (SVC), un descodificador de vídeo tal vez no pueda descodificar completamente múltiples capas de dependencia (con las mismas o diferentes resoluciones) debido al diseño de descodificación de un solo bucle del descodificador de vídeo. En este caso, cuando se utiliza un modo de capa de base (BL) intra-predicho, el descodificador de vídeo solo puede descodificar correctamente los MB de la capa de mejora si los MB intra de capa de base correspondiente se codifican como MB intra restringidos. De lo contrario, tal vez sea necesario que el descodificador de vídeo prediga los MB intra de capa de base de los MB inter próximos reconstruidos, los cuales tal vez no estén disponibles al descodificar la capa de mejora en caso de descodificación

de bucle único.

[0170] En respuesta a la determinación de que el píxel de borde se asocia con una CU inter-predicha y que la intra-predicción restringida está habilitada ("SÍ" de 458), el codificador de vídeo puede determinar que el píxel de borde no está disponible (454). Por otra parte, en respuesta a la determinación de que el píxel de borde no está asociado con una CU inter-predicha o que no está habilitada la intra-predicción restringida ("NO" de 458), el codificador de vídeo puede determinar que el píxel de borde está disponible (460).

[0171] En los ejemplos anteriores, un codificador de vídeo puede asignar un valor predeterminado o un valor de píxel de borde anterior a píxeles de borde no disponibles. En otros ejemplos, el codificador de vídeo puede interpolar valores de píxeles de borde no disponibles. Por ejemplo, si hay uno o más píxeles de borde disponibles en ambos lados de una serie de uno o más píxeles de borde no disponibles. En este caso, el codificador de vídeo puede interpolar los valores de los píxeles de borde no disponibles basándose en los valores de los píxeles de borde disponibles en cada lado de la serie de píxeles de borde no disponibles. Los píxeles de borde no disponibles pueden interpolarse como $U_i = A_a + i * (A_b - A_a)/(M+1)$, i de 1 a M . En esta ecuación, A_a y A_b son los píxeles de borde disponibles a cada lado de la serie de píxeles de borde no disponibles, M es el número de píxeles de borde no disponibles en la serie de píxeles de borde no disponibles y U_i indica los píxeles de borde no disponibles. En la ecuación anterior, M tal vez no sea una potencia de 2, por lo que puede ser necesaria la división.

[0172] Debido a que M tal vez no sea una potencia de 2 y puede ser deseable evitar la división, puede usarse otra técnica para interpolar la serie de píxeles de borde no disponibles. De acuerdo con esta técnica, sea $Q = \text{Floor}(\log_2(M))$, $P=(1<Q)$ y $R=M-P$. El codificador de vídeo puede establecer cada uno de los píxeles de borde no disponibles en la derecha, que son U_{P+1} a U_M , en el valor del valor de A_b . Para cualquier píxel de borde de U_1 a U_P ,

$$U_i = (A_a * i + A_b * (P-i) + P >> 1) >> Q, i \text{ de } 1 \text{ a } P.$$

[0173] En algunos casos no hay ningún borde disponible en ningún lado de una serie de uno o más píxeles de borde no disponibles. Por ejemplo, se pueden producir píxeles de borde consecutivos no disponibles hasta el final del predictor. En este ejemplo, el codificador de vídeo puede extender el valor del último píxel de borde disponible hasta el final del predictor. En otro ejemplo, se pueden producir píxeles de borde consecutivos no disponibles al inicio de un predictor. En este ejemplo, el codificador de vídeo puede utilizar el valor del primer píxel de borde disponible como el valor de los píxeles de borde no disponibles que preceden al primer píxel de borde disponible. Así, en tales casos, el codificador de vídeo puede usar una extensión simple para generar los valores para las muestras de referencia no disponibles.

[0174] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse, como una o más instrucciones o código, en un medio legible por ordenador o transmitirse a través de este, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder en general a (1) unos medios de almacenamiento legibles por ordenador tangibles que son no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0175] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse para almacenar un código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medios legibles por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota mediante un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. El término disco, tal como se utiliza en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, de los cuales el disco flexible normalmente reproduce datos de magnéticamente, mientras que el resto de discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior deberían incluirse también dentro del alcance de los medios legibles por

ordenador.

5 **[0176]** Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquiera de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la
10 descodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

15 **[0177]** Las técnicas de la esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, que incluyen un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Aunque en esta divulgación se describen varios componentes, módulos o unidades para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, estos no requieren necesariamente su realización mediante diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los
20 descritos anteriormente, conjuntamente con software y/o firmware adecuados.

REIVINDICACIONES

1. Un procedimiento para codificar datos de vídeo, comprendiendo el procedimiento:
 - 5 realizar una operación de relleno que procesa un conjunto de píxeles de borde, asociados con un bloque de vídeo actual, de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y continúa a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho, en el que cuando la operación de relleno procesa un píxel de borde no disponible que no sea el píxel de borde inferior izquierdo, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado mediante la operación de relleno; y
 - 10 generar un bloque de vídeo intra-predicho para dicho bloque de vídeo actual basado en los píxeles de borde procesados, **caracterizado por que** la operación de relleno asigna el valor al píxel de borde no disponible sin buscar un siguiente píxel de borde disponible en el orden de procesamiento.
- 15 2. El procedimiento según la reivindicación 1, en el que la realización de la operación de relleno comprende:
 - determinar si un píxel de borde actual está disponible; y
 - 20 asignar al píxel de borde actual un valor de un píxel de borde que está inmediatamente antes del píxel de borde actual de acuerdo con el orden.
- 25 3. El procedimiento según la reivindicación 2, en el que la determinación de si el píxel de borde actual está disponible comprende determinar que el píxel de borde actual no está disponible cuando el píxel de borde actual está asociado con una unidad de codificación (CU) inter-predicha (CU) y la intra-predicción restringida está habilitada.
- 30 4. El procedimiento según la reivindicación 1, en el que la realización de la operación de relleno comprende:
 - determinar si el píxel de borde inferior izquierdo está disponible; y
 - asignar un valor predeterminado al píxel de borde inferior izquierdo en respuesta a determinar que el píxel de borde inferior izquierdo no está disponible.
- 35 5. El procedimiento según la reivindicación 1, en el que la realización de la operación de relleno comprende:
 - determinar si el píxel de borde inferior izquierdo está disponible; y
 - 40 asignar un valor de un siguiente píxel de borde disponible al píxel de borde inferior izquierdo en respuesta a determinar que el píxel de borde inferior izquierdo no está disponible.
- 45 6. El procedimiento, de acuerdo con la reivindicación 1, en el que la realización de la operación de relleno comprende:
 - para $x = -1, y = nS*2-2 \dots - 1$, sustituir un valor de $p[x, y+1]$ por un valor de $p[x, y]$ si $p[x, y]$ es un píxel de borde no disponible; y
 - para $x = 0..nS*2-1, y-1$, sustituir un valor de $p[x-1, y]$ por un valor de $p[x, y]$ si $p[x, y]$ es un píxel de borde no disponible,
 - 50 en el que nS es igual a una anchura o altura del bloque de vídeo intra-predicho, p es una matriz bidimensional de muestras y $p[0,0]$ es una muestra de luma superior izquierda del bloque de vídeo intra-predicho.
- 55 7. El procedimiento según la reivindicación 1, en el que el procedimiento se realiza durante un proceso de codificación, comprendiendo además el procedimiento generar datos de vídeo codificados basados en el bloque de vídeo intra-predicho.
- 60 8. El procedimiento según la reivindicación 7, que comprende además generar datos residuales basados al menos en parte en el bloque de vídeo intra-predicho.
9. El procedimiento según la reivindicación 1, en el que el procedimiento se realiza durante un proceso de descodificación, comprendiendo además el procedimiento generar datos de vídeo descodificados basados en el bloque de vídeo intra-predicho.
- 65 10. El procedimiento según la reivindicación 9, que comprende además reconstruir un bloque de vídeo descodificado basado al menos en parte en el bloque de vídeo intra-predicho y un bloque de vídeo residual.

11. Un aparato de codificación de vídeo, que comprende:

5 medios para realizar una operación de relleno que procesa un conjunto de píxeles de borde, asociados con un bloque de vídeo actual, de acuerdo con un orden que comienza en un píxel de borde inferior izquierdo y continúa a través de los píxeles de borde secuencialmente hasta un píxel de borde superior derecho, en los que cuando la operación de relleno procesa un píxel de borde no disponible que no sea el píxel de borde inferior izquierdo, la operación de relleno asigna un valor al píxel de borde no disponible basándose en un valor de un píxel de borde previamente procesado por la operación de relleno; y

10 medios para generar un bloque de vídeo intra-predicho para dicho bloque de vídeo actual basado en los píxeles de borde procesados, **caracterizado por que** la operación de relleno asigna el valor al píxel de borde no disponible sin buscar un siguiente píxel de borde disponible en el orden de procesamiento.

- 15 12. El aparato de codificación de vídeo según la reivindicación 21, en el que los medios para llevar a cabo la operación de relleno comprenden:

medios para determinar si un píxel de borde actual está disponible; y

20 medios para asignar al píxel de borde actual un valor de un píxel de borde que está inmediatamente antes del píxel de borde actual de acuerdo con el orden.

- 25 13. El aparato de codificación de vídeo de la reivindicación 22, en el que los medios para determinar si el píxel de borde actual está disponible comprenden medios para determinar que el píxel de borde actual no está disponible cuando el píxel de borde actual está asociado con una unidad de codificación (CU) inter-predicha y la intra-predicción restringida está habilitada.

- 30 14. Un producto de programa informático que comprende uno o más medios de almacenamiento legibles por ordenador que almacenan instrucciones ejecutables por ordenador que, cuando se ejecutan, hacen que uno o más procesadores realicen el procedimiento de cualquiera de las reivindicaciones 1 a 10.

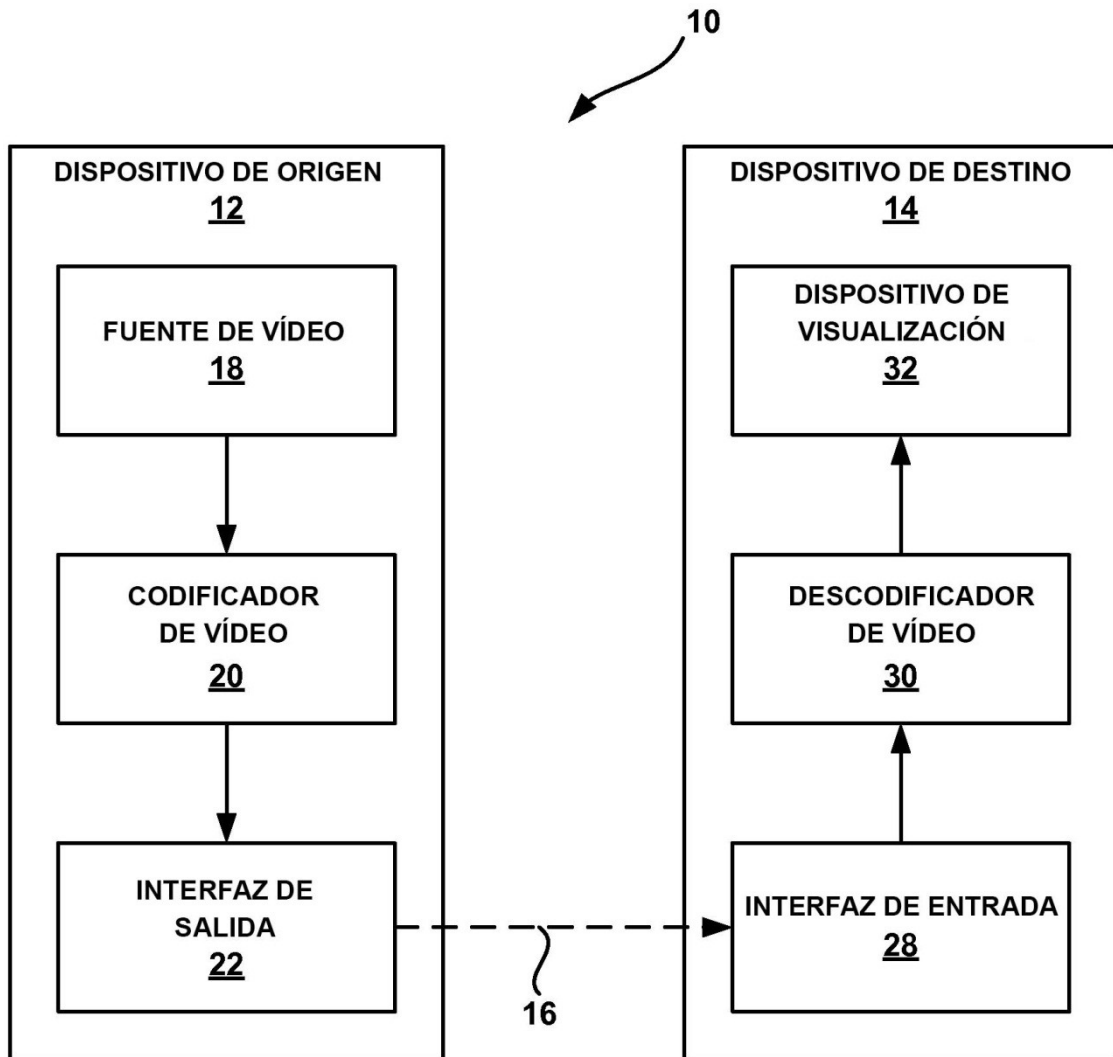


FIG. 1

R	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				
M								
N								
O								
P								

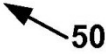


FIG. 2

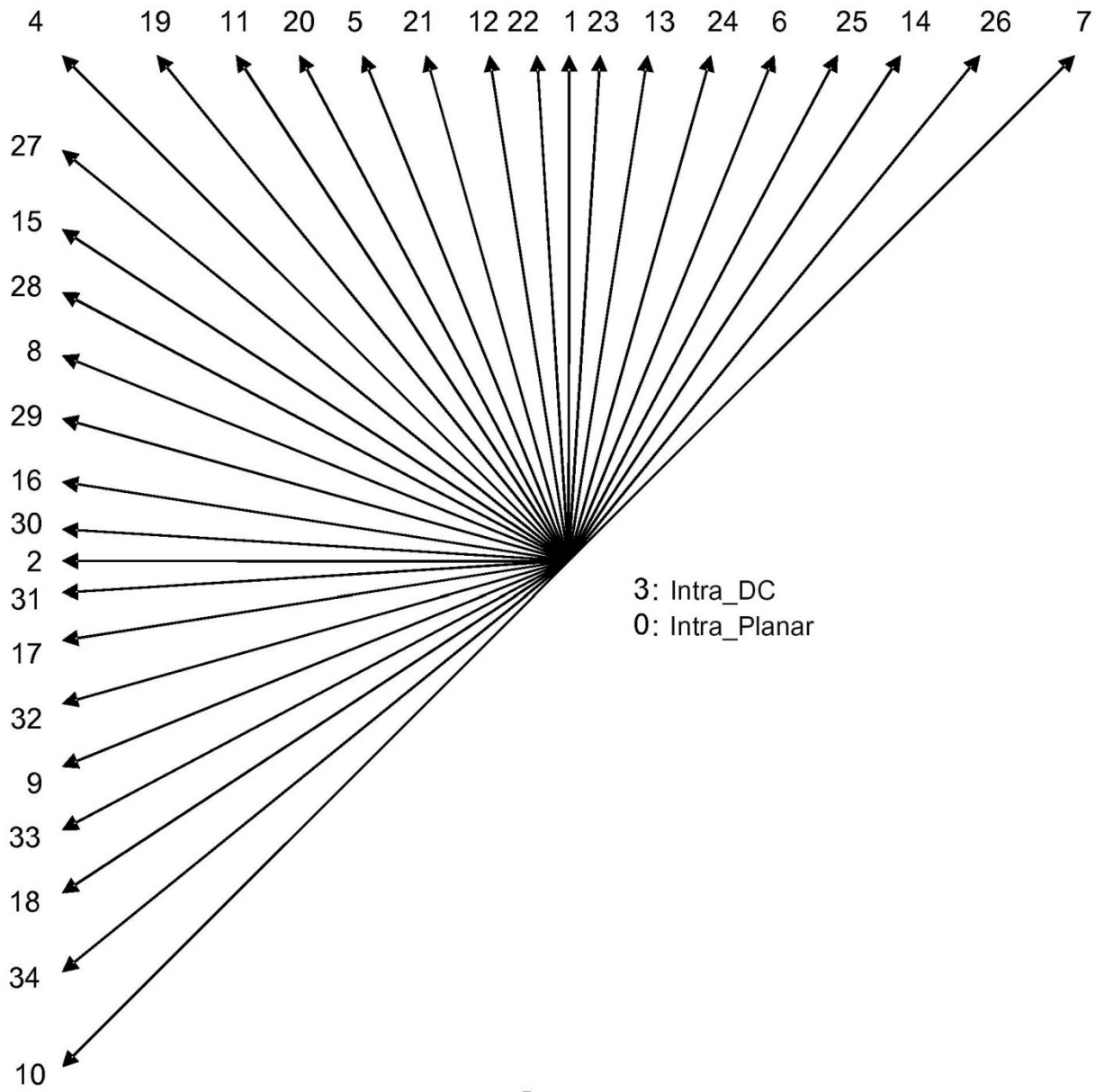


FIG. 3A

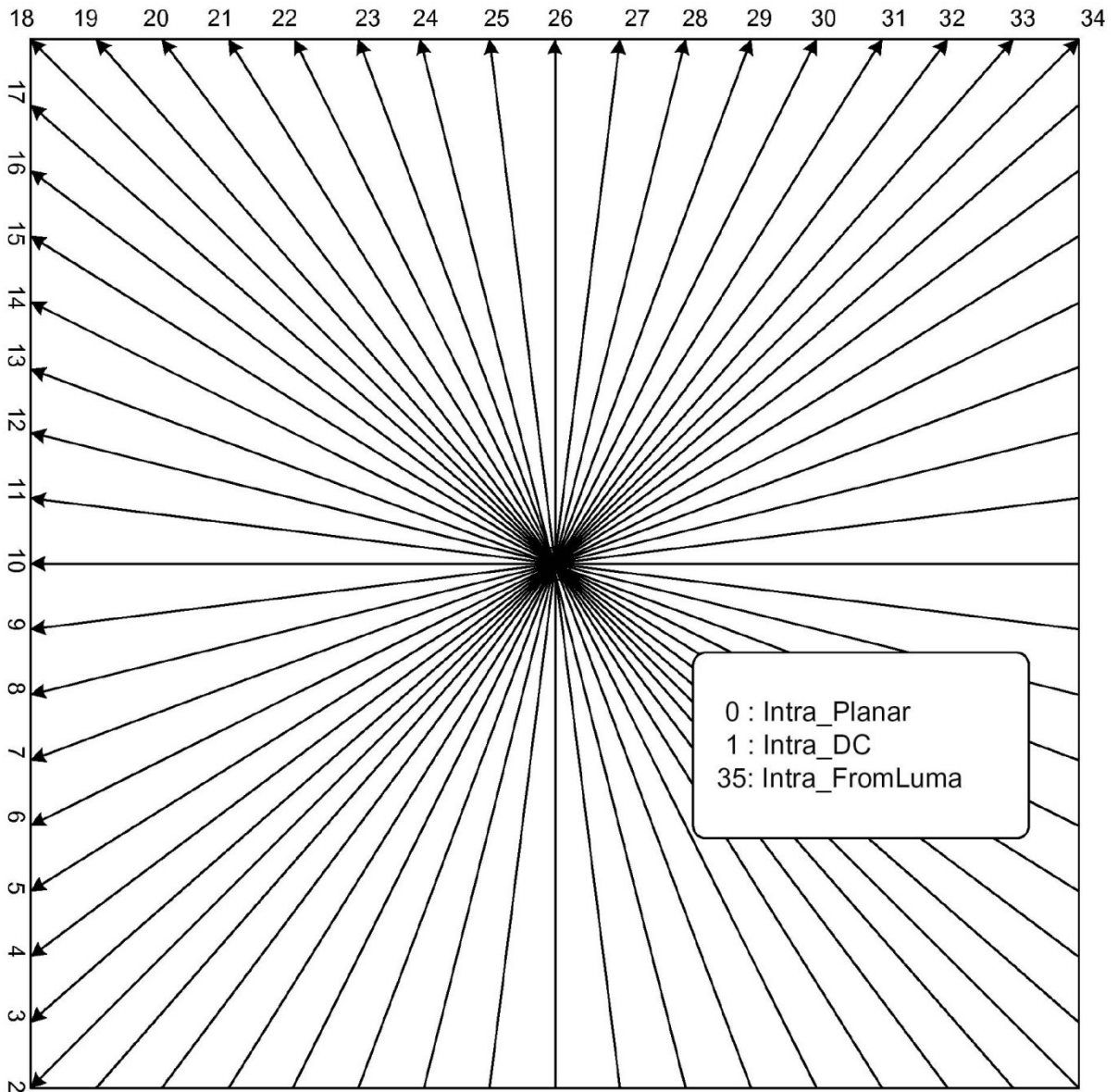


FIG. 3B

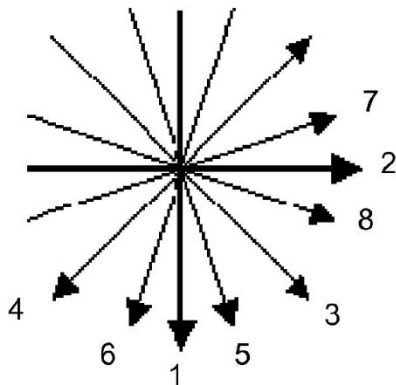


FIG. 3C

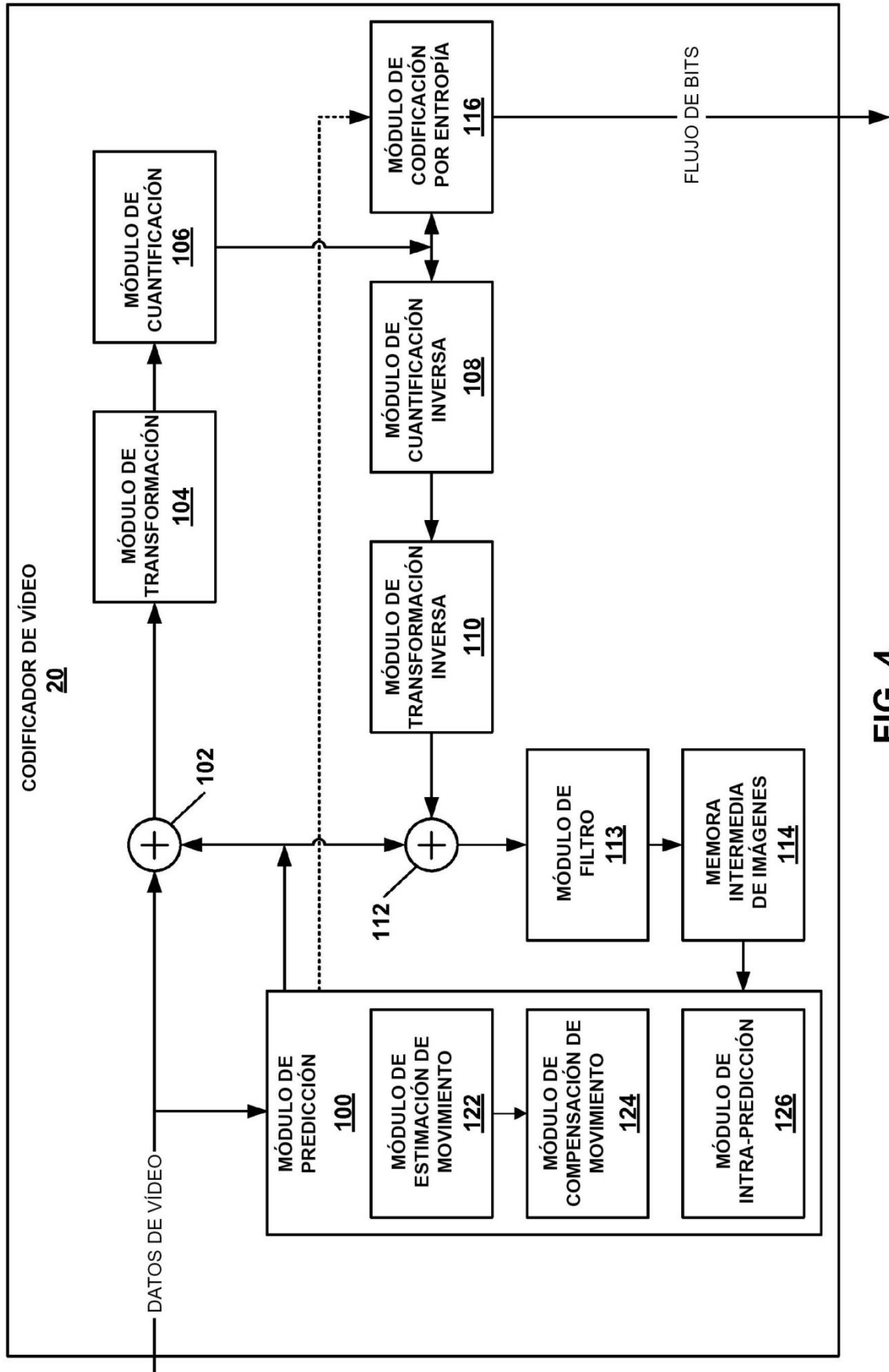


FIG. 4

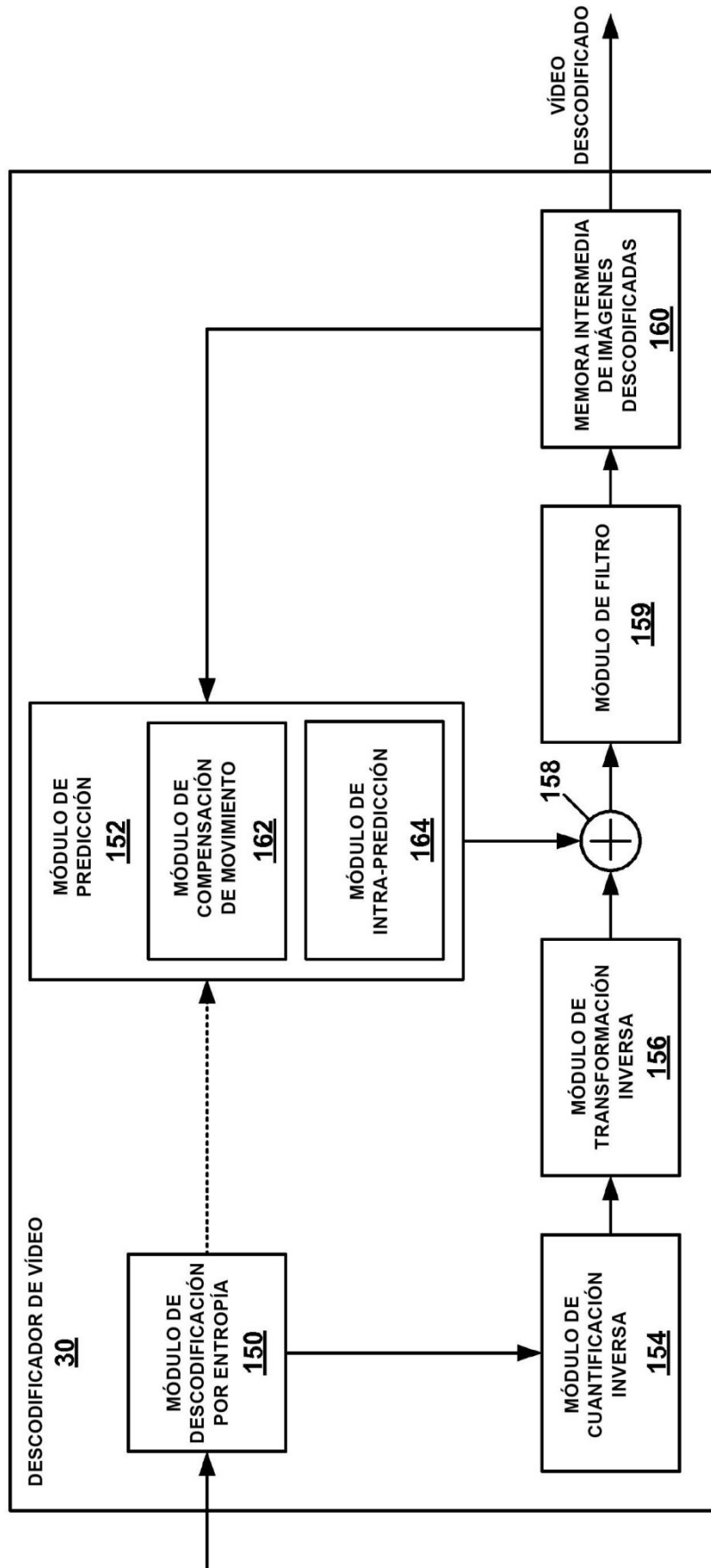


FIG. 5

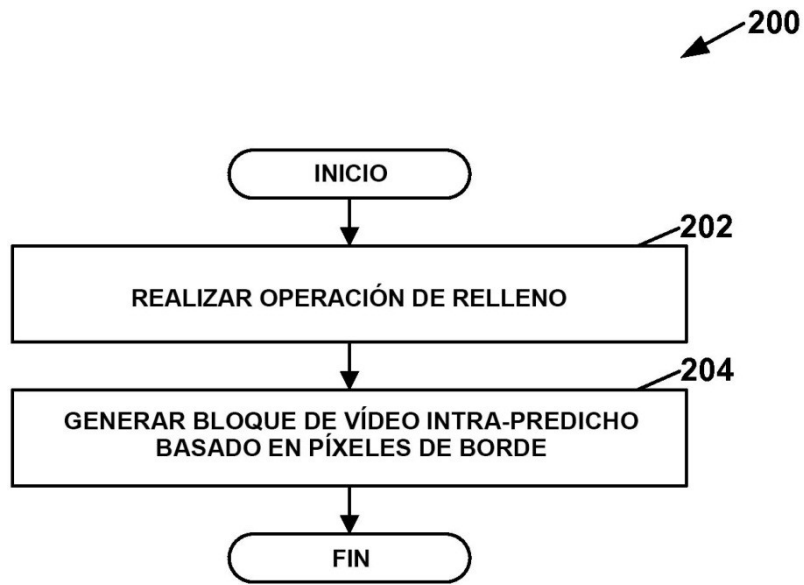


FIG. 6

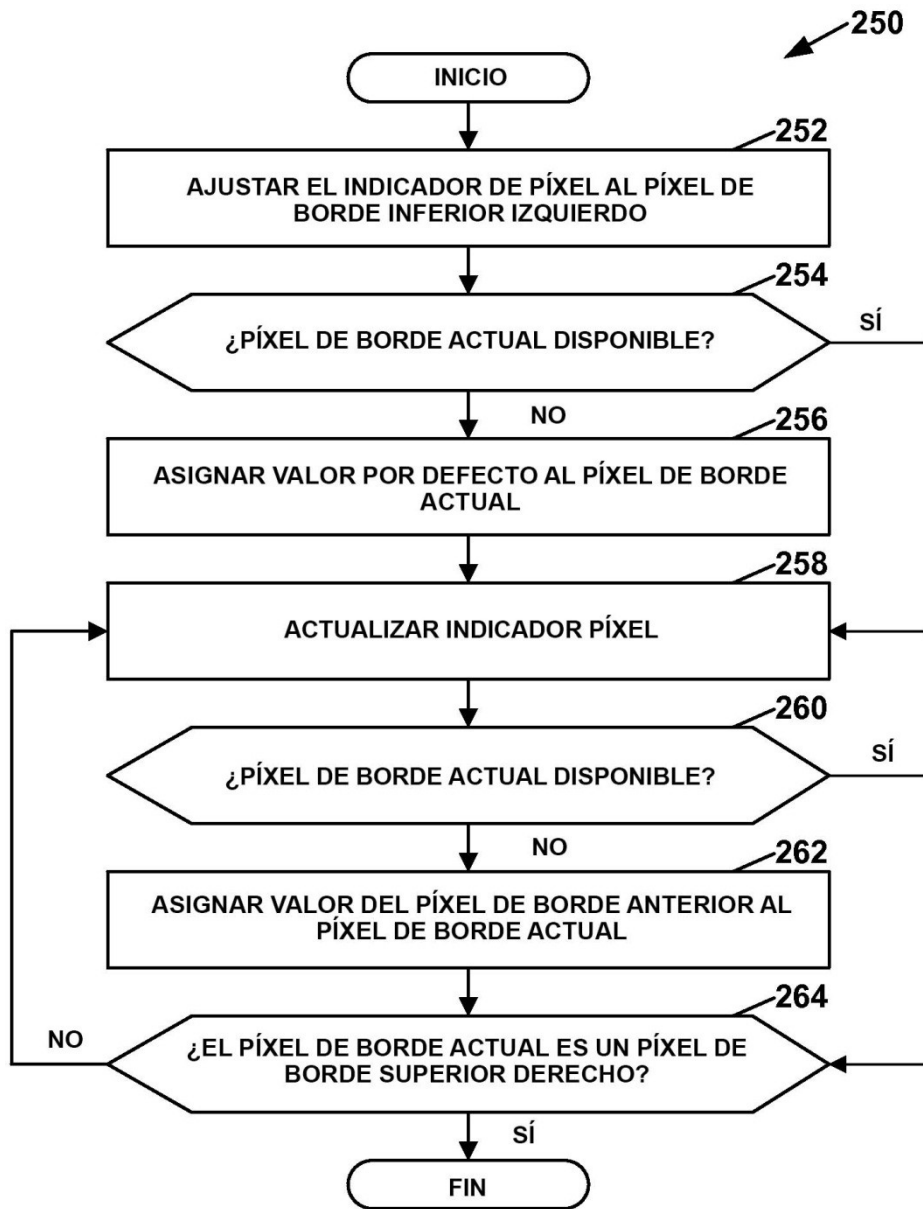


FIG. 7

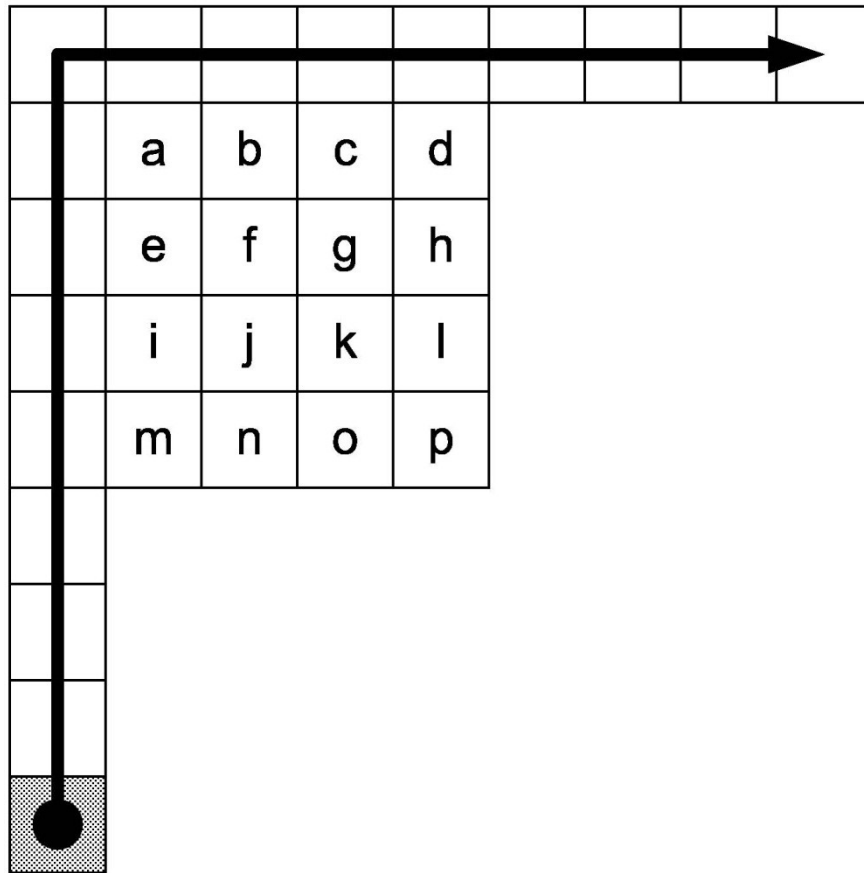


FIG. 8

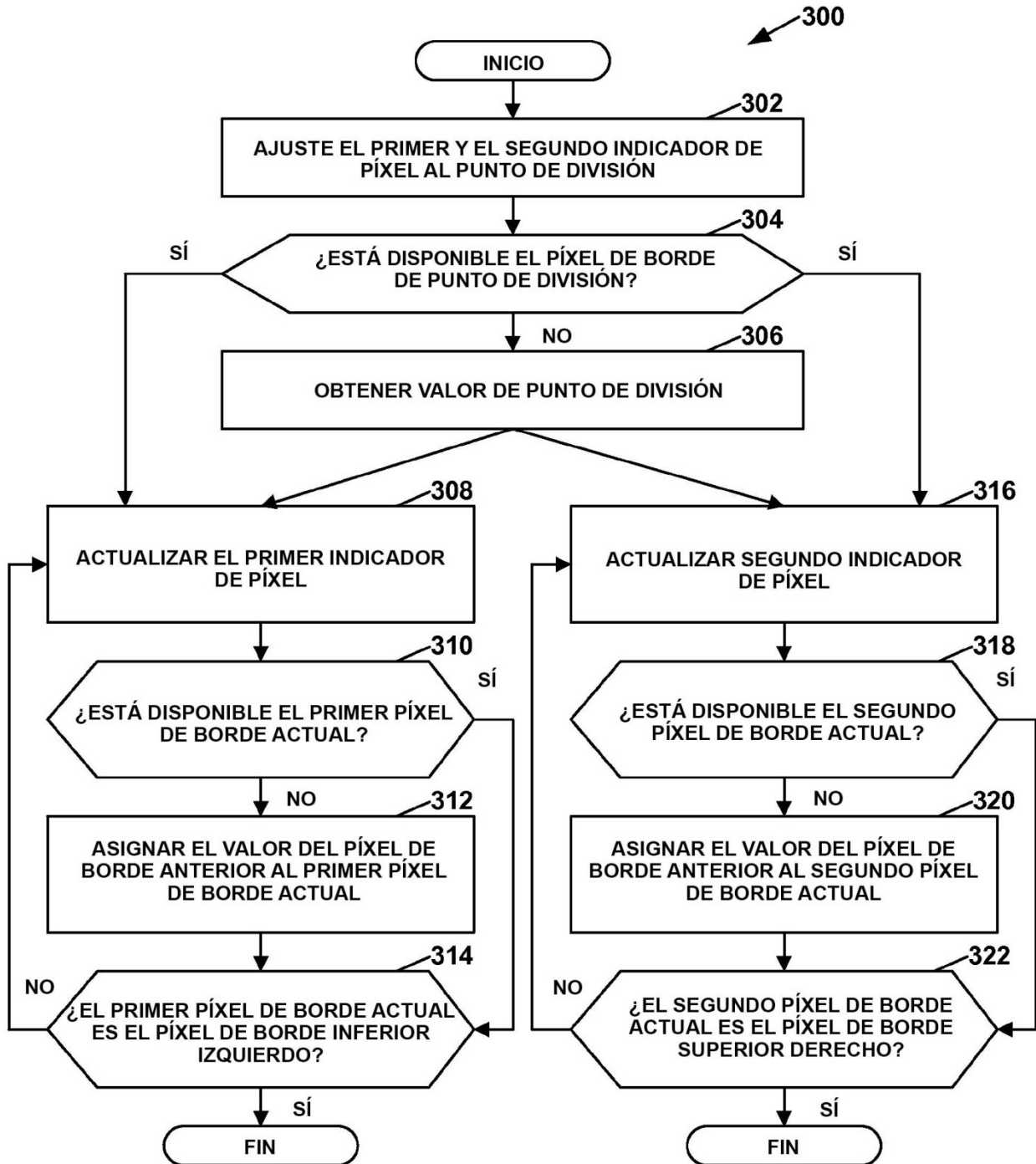


FIG. 9

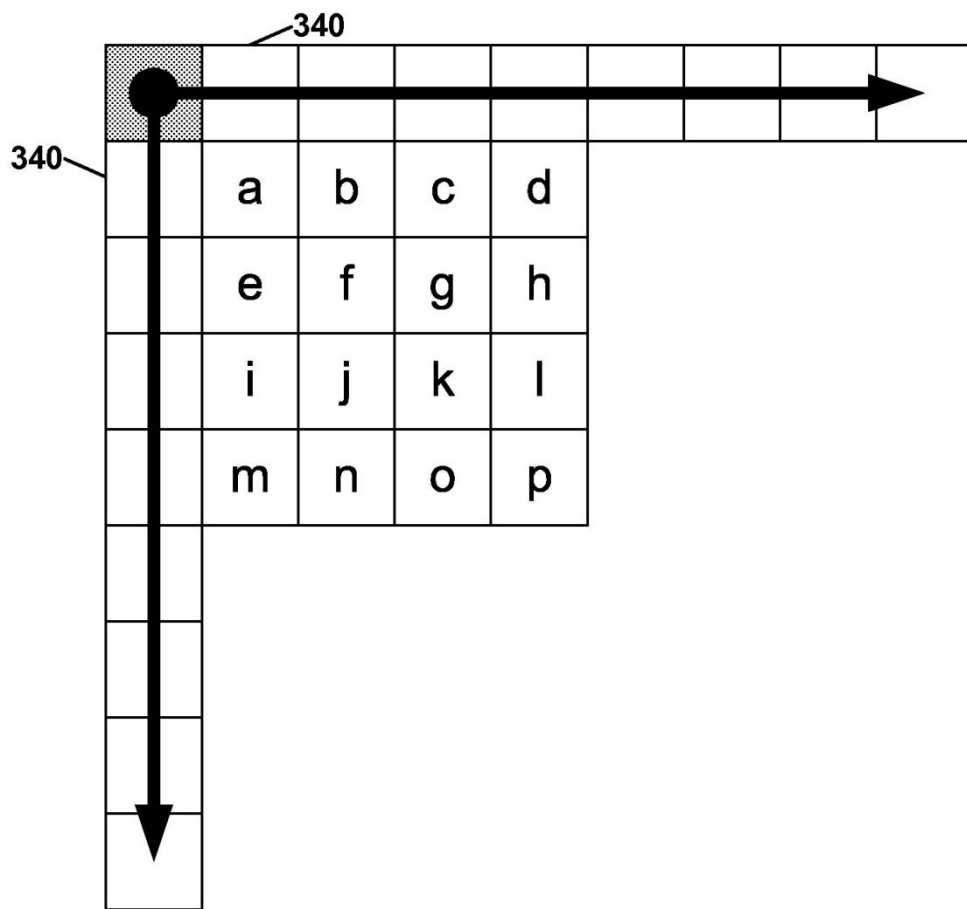


FIG. 10

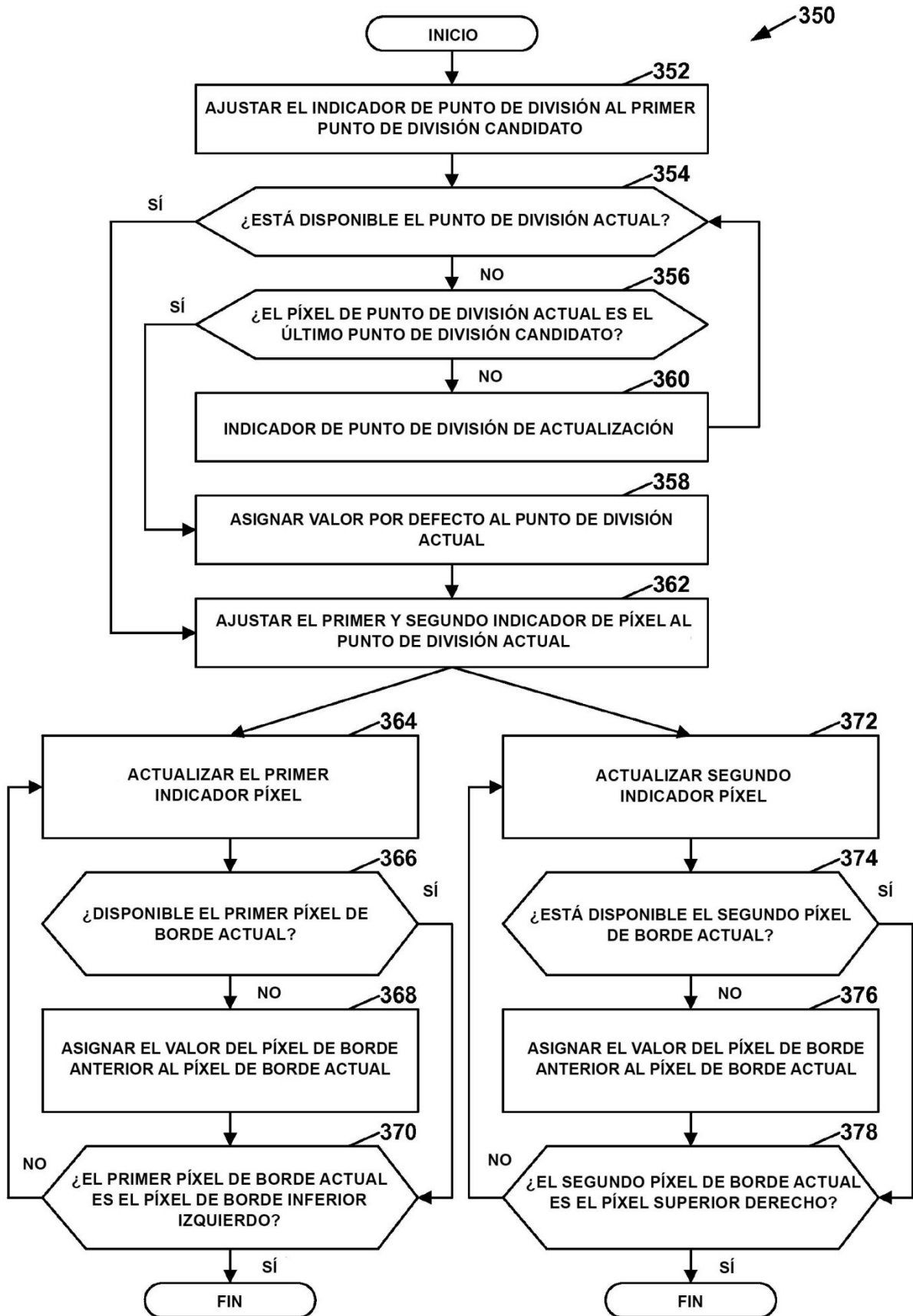


FIG. 11

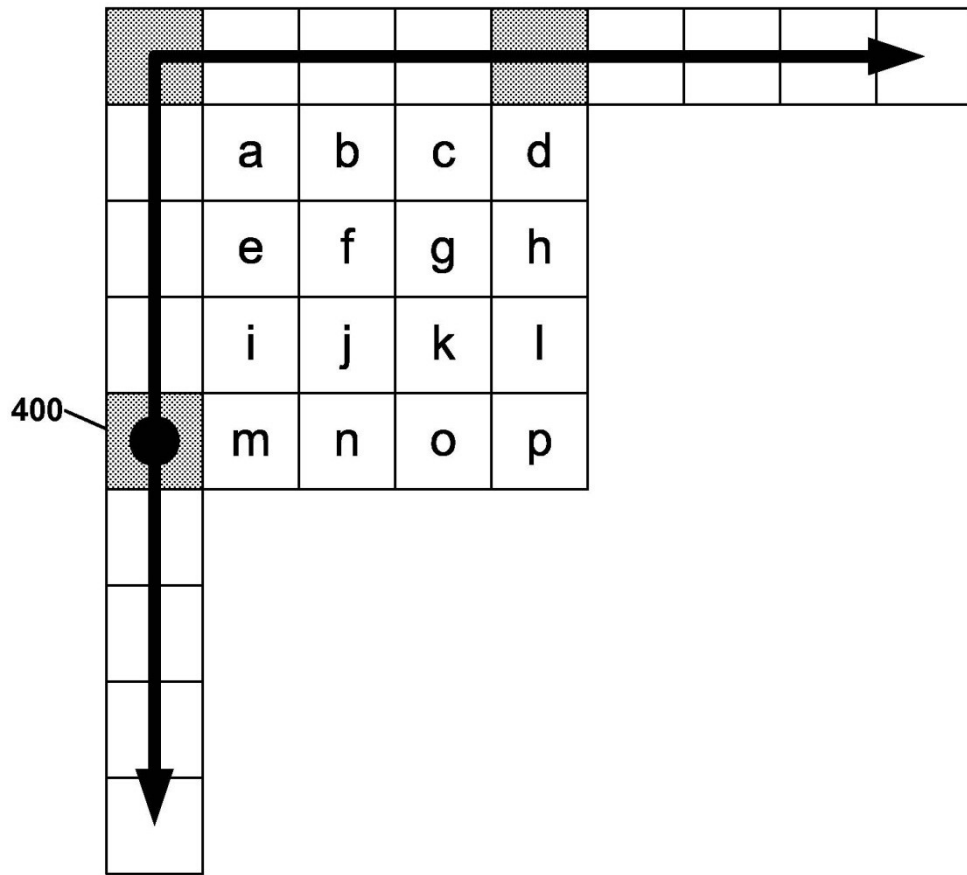


FIG. 12

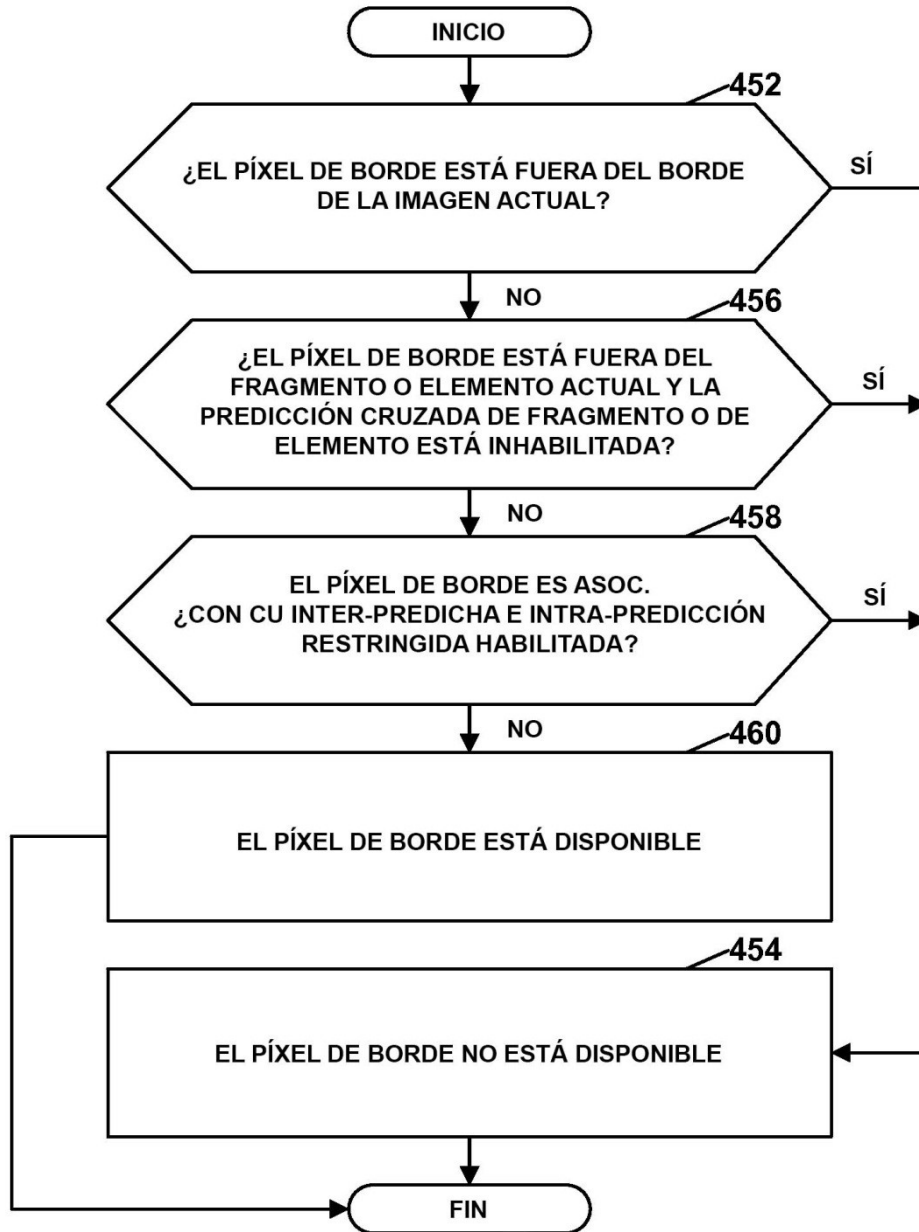


FIG. 13